UNIVERSITA' DEGLI STUDI
DI MILANO
DIPARTIMENTO DI SCIENZE
DELL'INFORMAZIONE

BULL ITALIA
CENTRO STUDI
INFORMATICA
E AUTOMAZIONE

Note di Software

ARCHIVIO
STORICO
ASSOCIAZIONE
POZZO DI MIELE
FPDM-125
RNSW-133

Anno 1990
48/49
Giugno/Ottobre

Note di software è una pubblicazione trimestrale, edita a cura del Dipartimento di Scienze dell' Informazione dell' Università di Milano e dal Centro Studi Informatica e Automazione della Bull Italia.

Note di software intende costituire uno strumento per la circolazione di informazioni, idee, metodologie ed esperienze nell'area delle tecnologie e della produzione del software.

La collaborazione a **Note di software** è libera. Chi desiderasse pubblicare un contributo è pregato di prendere contatto con il Direttore responsabile o con il Comitato di Redazione. In particolare si sollecitano contributi nella forma di monografie, da inviare in triplice copia, e che non dovrebbero superare le cinquemila parole.
I lavori ricevuti per la pubblicazione verranno revisionati dai membri del Comitato di Revisione Scientifica, che possono avvalersi della collaborazione di specialisti del settore. I lavori pubblicati riflettono comunque le opinioni dei loro autori.

Note di software viene distribuito ad Aziende, Enti e specialisti del settore che ne facciano richiesta alla Redazione.

Direttore Responsabile: Franco Filippazzi
Bull Italia, Centro Studi Informatica e Automazione, via Vida 11 - 20127 Milano

Comitato di Redazione: Stefania Bandini, Francesco Gardin, Roberto Polillo
Università di Milano, Dipartimento di Scienze dell' Informazione
via Moretto da Brescia, 9 - 20133 Milano, tel. (02) 7575233

Comitato di Revisione Scientifica: Alberto Bertoni, Gianluigi Castelli, Giovanni Degli Antoni, Giorgio De Michelis, Mario Italiani, Gaetano Aurelio Lanzarone, Giancarlo Martella, Marco Maiocchi, Giancarlo Mauri.

Autorizzazione del Tribunale di
Milano n. 87 del 23 febbraio 1977

Supporto tecnico - gestionale e distribuzione: Domenico Asinelli
Bull Italia, via Tazzoli 6
20154 Milano, Tel. 02 - 67792960

UNIVERSITA' DEGLI STUDI
DI MILANO
DIPARTIMENTO DI SCIENZE
DELL'INFORMAZIONE

FRDM-125

Bull Italia
CENTRO STUDI
INFORMATICA
E AUTOMAZIONE

ABOUT THIS SPECIAL ISSUE 1

- GRAPHICAL INTERFACES, HYPERMEDIA, COMMUNICATION, SOFTWARE TECHNOLOGY
  Gianni Degli Antoni 3
- INTERACTING WITH ARTIFICIAL REALITIES
  Roberto Polillo 7
- A MODEL FOR ARTIFICIAL REALITY
  Francesco Gardin 29
- SYMMETRY IN MAN-MACHINE COMMUNICATION
  Roberto Grande 38
- SELF-OBSERVATION IN ARTIFICIAL REALITY
  Rita Pizzi 46
- HYPERMILAN
  Marina Bonomi, Marcella Logli 51
- THE CONCEPT OF REUSABILITY IN EDUCATIONAL DESIGN
  Giorgio Olimpo 63
- SUPER-LIBRARY: AN ENVIRONMENT FOR THE CONVERSION OF TEXT INTO HYPERTEXT
  Flavio Argentesi, Nadia Catenazzi 69
- HYPERCHIP: A SYSTEM ENGINEERING ENVIRONMENT WITH HYPERTEXT FRONT-END
  Stefano Antoniazzi, Mirella Mastretti 76
- HIGRAPHS FOR HYPERTEXT MODELLING
  Carlo Alberto Bosello, Fabio Vitali 84
- HYPERNET
  Marco Bondi, Maurice Pagnucco 98
- A HYPERMEDIA INFORMATION SYSTEM TO MANAGE THE ACTIVITIES IN A RESEARCH INSTITUTE
  Mario Allegra, Onofrio Di Giuseppe, Silvana Mangiaracina 115
- "THE HEART: ITS FUNCTIONING AND DISFUNCTIONING": LESSONS LEARNED AND OPEN QUESTIONS FROM A CD-ROM PROJECT
  Alessandro Campana, Riccardo Cattaneo, Patrizia Ghislandi 124
- HYPERMEDIA SECRETARY FOR DOCUMENT MANAGEMENT
  Piera Piardi 138
- VISUAL-SYMBOLIC INTEGRATION IN NEURORADIOLOGY
  Francesco Bonadonna 155

RESEARCH NOTES AND COMMUNICATIONS

- DATA GLOVE AND VISUAL ART
  Adriano Abbado 164
- FROM TEXT TO HYPERTEXT THROUGH METATEXT
  Armando Limongiello, Filippo Soresi, Andrea Spinelli, Walter Vannini..... 168
- BROWSING IN KNOWLEDGE THROUGH ANIMATION: AN APPLICATION TO THE A.I.D.S. CASE
  Mario di Floriano 170
- HYPERMEDIA AND BLIND PEOPLE TODAY
  Nunzio La Ferlita 174
- A PROJECT ON SOUNDS IN ARTIFICIAL REALITIES
  Francesco Gardin, Goffredo, Haus, Ilaria Macchia, Gloria Molendi 177

Note di Software

Anno 1990
48/49
Giugno/Ottobre

SPECIAL ISSUE ON MULTIMEDIA AND ARTIFICIAL REALITY

The growing interest in and importance of multimedia technologies and Artificial Reality, has indicated the need for a special issue on the subject. In order to increase the circulation of this issue, invited authors have been asked to send their contributions in English.
The publication of this issue also coincides with the opening of the Conference "Palermo 90: Sistemi Multimediali e loro Applicazioni", co-sponsored by the Italian Research Council. This year's topic focuses on Multimedia Technology for Teaching. Two Conference papers have also been included in this Special Issue because of their particular pertinence and as representatives of the quality of the contributions presented at the Conference.

What follows is a short description of the contents of each article, to give the reader a concise overview of the topics presented by the authors.

The Special Issue opens with an article by Gianni Degli Antoni. After some general remarks about current software development technology and the software industry, especially in relation to recent trends in OOP's, graphical interfaces and hypermedia, he presents a prototyping approach to software development, centered on the notion of realistic prototyping.

The next paper, by Roberto Polillo, is an overview of advanced man-machine interfaces as primary examples of interaction with Artificial Reality, and discusses how the notion of man-machine interaction will change in view of this trend.

Francesco Gardin raises the issue of the need for a model for Artificial Reality and its design principles. A solution, based on the notion of "hard" and "soft" agents, and how they can be combined together, is presented together with a list of design principles and examples.

The contribution by Roberto Grande is about the problem of symmetry in communication between man and computer, described in syntactic, semantic and pragmatic terms. The use of the ProTalk system of a common language and model of pragmatics is described.

In her paper, Rita Pizzi describes a formal model of the infinite levels of cognitive reflexion caused by interaction with Artificial Reality. This results in an "equation of Artificial reality".

The paper by Marina Bonomi and Marcella Logli contains a detailed report on the Hypermilan project: a multimedia information system about Milan.

Giorgio Olimpo raises the issue of the reusability of multimedia courseware and lists some guidelines for its production which should make reusability possible.

In their article, Flavio Argentesi and Nadia Catenazzi address the problem of converting existing paper documents into hypertext. Super-Library, the conversion environment developed for this purpose, is described.

Stefano Antoniazzi and Mirella Mastretti describe Hyperchip, the prototype of an interactive system which uses a hypertext as the navigation metaphor in a hyperspace of representations for the design phase of hardware and software systems.

Carlo Alberto Bosello and Fabio Vitali give an account of RHYTHM, a research project with the aim of defining a general mathematical structure of different hypertexts, based on the notion of "higraph" and its "s-higraphs" extension.

The article by Marco Bondi and Maurice Pagnucco presents Hypernet, a multimedia system based on Petri nets, conceived as a representation, analysis and design environment for systems in general.

The contribution by Mario Allegra, Onofrio Di Giuseppe and Silvana Mangiaracina, also included in the Mediterranean Conference Proceedings, describes a hypermedia application developed to assist the management and support of research activities in a Research Institute.

In their paper Alessandro Campana, Riccardo Cattaneo and Patrizia Ghislandi write about the project for the development of the hypermedia CD-ROM "The heart: its functions and disfunctions". The solutions identified to overcome some of the problems encountered during the design and implementation phases of the project are presented.

The S2 prototype which Piera Piardi describes in her article is a system based on hypermedia technology, conceived as an interface to an Office Automation environment, where specialized agents take care of the execution of specific tasks.

Francesco Bonadonna, also included in the Mediterranean Conference Proceedings, ends the paper session of this Special Issue. The author presents a new theoretical framework for representing the integration of images and text in Radiology, using image database structures and hypermedia.

The RESEARCH NOTES AND COMMUNICATIONS section groups together a number of contributions which are short notes about ongoing research and cultural activities related to the main topics of this Special Issue.

Adriano Abbado reports on his performance, as an artist, at the "Nuove Immagini e Formazione" meeting, held in Venice last December, where, using a Data Glove to control computer-generated images and sound, he achieved a new level of interaction.

Armando Lomongiello, Filippo Soresi, Andrea Spinelli and Walter Vannini in their communication present the concept of metatext, as a new level of description in a document of conceptual structures such as notes, references, glossaries, etc.

Mario di Floriano presents an interactive multimedia prototype for accessing and updating a knowledge base related to AIDS.

In his note, Nunzio La Ferlita deals with the issue of blind people performing ordinary activities and how hypertext and multimedia are failing to accomodate their needs even though they could have a fundamental role in reducing the effects of this handicap.

Lastly, in their research note, Francesco Gardin, Goffredo Haus, Ilaria Macchia and Gloria Molendi give an account of the state of the "SOUNDS WORLD" project which centres on the introduction of a sound model into Artificial Reality.

"Note di Software" wishes to thank Roberta Macchi, for her contribution as co-editor for this special issue and her work with the Hypertext User Group (HUG) at the Dipartimento di Scienze dell'Informazione (DSI) of Milan University. Finally, special thanks to Prof. Gianni Degli Antoni, DSI, who has been the driving and inspiring force behind HUG, Artificial Reality and most of the Department's multimedia research projects.

THE EDITORIAL BOARD

GRAPHICAL INTERFACES, HYPERMEDIA, COMMUNICATION, SOFTWARE TECHNOLOGY

Giovanni Degli Antoni
Dipartimento di Scienze della Informazione
Università degli Studi
Milano

RIASSUNTO

La nota presenta una rassegna compatta dello stato dell'arte nello sviluppo software in relazione alla introduzione di tecnologie ad oggetti, alle interfacce grafiche ed ai concetti della ipermedialità.
Successivamente viene presentato l'obiettivo principale dello scritto: una metodologia di sviluppo software basata sulla prototipazione, che sottolinei già nelle prime fasi l'ottenimento di prototipi realistici direttamente utilizzabili.

ABSTRACT

The paper presents a compact overview of the actual state of the art in software development, particularly in relation to the introduction of object oriented technologies, graphical interfaces and hypermedia concepts.
The main objective of the paper is then presented: a software methodology development based on prototypes where a strong attention is placed on obtaining realistic spin-offs even from the earliest stages of the development.

1. THE CONTEXT

The real world is demanding easily used system development from the computer community in order to enable people to extend their use of computers to new subjects, as well as to improve organization and the quality and productivity of people's work. Optical memories are offering the possibility of collecting together as many documents necessary in any activity for recalling methods or data. Professional work stations are becoming both cheaper and more powerful. World competition is increasing as a consequence of the hopefully pacific emergence of many new countries that will compete in information technology. Those on the selling end are realizing that software represents a challenging opportunity. Meanwhile every one of us has understood that software is a rather variegated field whereby different attitudes are taking hold as a consequence of various softlogic (as analog to geologic) epochs concurrently surviving in the same place, frequently in the same organization. Artificial intelligence is spreading, taking the shape of expert systems in many fields helping to make decisions. Increasing labor costs suggest that development of software should be minimized to real needs: packages are more and more preferred in connection with personal workstations. Workstations are becoming the connection to main frames, and these are becoming a switching or main store system. Local area networks are widespread. Efforts to integrate every thing move ahead at a fast rate, consistent with the rather limited technological possibilities and with the high cost of integration in relation to

its advantages. Organizations are becoming increasingly aware that information systems development requires engineering technologies to put things together in an evolutionary system. Security and reliability are becoming a red lamp in the field. Viruses are teaching the weakness of networks and systems. The comparison and synergy between big and small are changing: they are becoming a comparison and synergy between very big and a very big small. Those on the selling end are not very happy with the market. Users want evident advantages and begin to dislike words such as performance, architecture, byte, and virtually every technical work. Research frequently disregards technology as if it were a non computing reality, perhaps as consequence of the fast evolution. Low end software people are slowly disappearing. Medium software people are becoming artisans and are going at the low end output. High end competent people are realizing their inadeguacy in extending their work to the needs; they are also too few in number. Standards are slowing down every thing and complicating competition. The legal aspects of successful software are making software production and commercialization a rather legal activity where creative people, architects, programmers, users and lawyers work together.

Commercialization is showing the tremendous difficulties in selling and protecting. The diffusion of software has until now not taken its right path. Advertizing is sometime nice to see, but scarcely informative. Computer shops sell software as a black box with a name, when it is known. Good people from universities do not always find the desired work. Industries do not always like high end people. 'Low cost offer work' in software developement is appearing from countries near and far, and some one is exploiting that offer to gain competitivity. High end software development is sometimes funded by research organizations. The ability to transfer research into a product is rather limited, at least in Europe. New technologies are appearing. In the software field these are: object oriented technologies, cooperative computing, communication between applications, cooperative work, hypermedia, hypertexts, mutimedia. New markets are developing in the field of electronic publishing, with extraordinary new products that have begun to understand the real possibility of integrating, many until now, different cultural as well as technological entities. Information integration is beginning to show that the idea of worldwide literary documentation centers and the distribution of large amounts of documents through optical memories is an actual reality. Paper is extending its power everywhere, but the knowledge worker is devastated by the amount of paper. Bulletin boards and specialized data banks are not a solution: they are becoming too many. Shows and conferences create much confusion hoping to sell more ideas and products and in order to advertize something, directly or indirectly. Schools at every level are no longer able to synchronize with real needs. Many organizations are creating a new large market trying to solve the teaching problem. Unfortunately, teaching is not always geared in the right direction. Teleteaching and new instructional technologies seem to have improved in quality and market. The demand for formal treatment of things is improving, while the formal systems ability to handle the new complex entities is rapidly decreasing.

2. SURVIVAL

Certainly guidelines for survival (surviving) are required. The problems to be faced are rather difficult because of the contradictory effect of the various solutions which could eventually be adopted. In many cases, for example, adopting optimization produces positive effects in one component, but frequently has a totally negative effect in another component: the overall is unpredictable. And all the components depend unavoidably on the all. So, optimization has to be adopted but concurrently with extensions of applications and solving the increased demand of integration and quality in information services that every information entity has to give. In this way the market will not be limited and a new dynamic of change could give rise to new growth. Who will lead such a difficult change is not clear at all. Everyone who has an interest in computing, in the quality of life and in society or simply in his/her company or his/her personal position, can make a contribution. The quality of systems and services together with an easy integration is probably the key point. And quality also means easy usage: without easy usage the information technology will reach no-one else, and this will result in a consistent slow fall of growth. There are many ways to obtain these results technically, even without sacrificing the sometimes old main frame. It is enough to interface it with brilliant personal work

stations. But if teaching is needed it has to be integrated into the work place, trying to avoid forgetting the great importance of user satisfaction for teaching technology. But this is not enough: new lines of information services have to be considered: cooperative work, integrated documentaion for the organizational know-how management, selective dissemination of information, a telematic view of the organization for its users, active catalog of products to be distributed (and sold) as products for the custom world, executive information systems, CAD of every kind (organization cad, building cad, plant cad, work cad, transportation cad, energy consumption cad, environment impact cad, financial cad, planning the future cad, marketing cad, data analysis cad, presentation cad....). But what really matters is that all the new services be designed for the final user: the employer or the medium level manager. And the user himself must be able to integrate everything to build his own applications.

3. PHILOSOPHICAL VIEWS

Good products survive. Product design depends on our ability to improve products or at least to understand what is the best development direction taken by high quality designers. To understand quality in design, a change in custom has to be accomplished: it is no longer certain which company products are the best (this is true for all companies!). And it is somewhat irresponsible to follow management indications without proposing examples for better solutions. The claim that the market is the guideline is correct in many cases, but not during difficult times. Moreover, market analysis has to be made with reference to all companies, and with the clear idea that small things grow. Big things not rarely get smaller.

So a unifying view of technology has to be given to make the direction of computer technology ideas understandable, and in order to synchronize choices in that direction. The view is rather obvious: information technologies must be considered the natural extension of the evolution of human communication. Many things have been proposed in the past for improving human communication in time and space. Data recording and data transmission systems were invented. Now it is time for the role of the computer in human communication (not in data communication or data transmission, even if without good data communication a good human communication will be impossibile) to emerge strongly. The role of computing in communication has many facets:

-the computer makes it possibile to reuse competence written in software or knowledge based entities without knowing the details of the competence, but by using the results of computations

-with computers it is possible to communicate in presence of an artificial context in order to make the recall of meaning to be associated to the communicated signs easier. This latter possibility is related to the use of voice, texts, images and software execution, all of these associated with an improved man-machine interface implementing an artifical context as a spatio temporal locus of communication. This 'spatio temporal' context creates new artificial or virtual realities. In the case of virtual realities the user of these new systems will perceive organizations or real entities exactly as the abstraction considered makes possible. Virtual machinery was once the first technical suggestion. Now we have to consider virtual plants, virtual offices, virtual organizations, virtual homes, virtual environments, virtual lecture roooms, virtual design labs..... And if nothing corresponds to the world evoked by the computer interactions, purely artificial realities will be displayed. The artificial realities could be future buildings / offices/ organizations/ plants, knowledge worlds, future systems of any kind, even pieces of the world of ideas. Plato called the world of ideas IPERURANIA. From this perspective it is quite correct to call the systems we are drawing as hypermedia systems, even if the more common name multimedia sytems is not inappropriate.

4. REALISTIC PROTOTYPING

The above picture has some advantages: to design a hypermedia system is a clearly understandable activity. Systems have to modeled. Interactions with those systems have to specified. Documentation accessible starting from the various points of a system has to be selected. Purposes related to the many parts of the system have to be displayed with the parts (let's call this: in context structural analysis). Properties have to be entered. Locations for the change of parts have to be predicted..... In short: the system structure helps in effecting analysis and design, helps in taking account of

parts, helps in visualizing problems and solutions...From a technical point of view, what we said is just an object-oriented analysis and design. And this approach makes the building up of complex systems more feasible where many entities communicate by means of messages. But, moreover, an object-oriented approach will make it feasible to design systems whereby the choices have to be made in suitable palettes (or classes). In this way it will be possible even for a causal user to build up extremely complex entities. This simplifying view of system development is considered essentially for the final user. For the software technologist the effort of developing something significant is becoming more and more complex and difficult. The decreasing distance between a technology and its successors tends, moreover, to raise the cost of developments due to the many changes introduced during development.

A development strategy is required. There are many suggestions. These have to cope with specifications, prototyping, testing, evalution specifications and performances, evaluating quality and user interface, selecting a continuously changing software/ hardware platform and standard... Frankly, if something sometimes happens it has to be considered a miracle. There are too many variables to take into account. Nevertheless, increasingly better products are appearing. The obvious thought of using extremely competent people is not an idea.

The following is a proposal.

Z) Select the medium long range goals of the project as advanced as you can imagine. Call it UTOPIA. Define it in a very short document clearly written. Ambiguity has not to be excluded.

This is not a simple task: it requires looking outside your views. But it has the significant advantage that you must look at the best competitors, small or big. But please, avoid underestimating the existing solutions, and do not forget that somebody else is working for the future...

A0) To start the above definition, build up the best approximation of your UTOPIA using the best hardware and software platforms, the best similar packages, and writing the minimum possible amount of software. Call it the zero realistic prototype.

The advantages are rather evident: it is easy to convince anybody of what you want to do. Integration problems will not all be solved. You will have the results in a shorter time than compared with other methods. You will be competent, in terms of competition and related products. Difficult specification writing will be fully avoided. User can be introduced in the evaluation at an early stage. Testing and performance evalutaion can be done immediately. The evalutaion of feasibility with other platforms or technologies of the prototype will be quite simple for an expert. But you require such an expert for a limited time. The you have to decide what to do.

Ak) If the zero realistic prototype does not match the utopia enough, or if you have not changed idea, then you have to take a sequence of approximations leaning towards UTOPIA.

At each step (internal releases) advances will be obtained, documentation will be produced in relationship with the foregoing prototypes and in relation with further functionality or other improvements (performance, software or hardware substitution, better user interface..). When satisfied, you have to stop, perhaps never, because intermediate steps of the sequence towards UTOPIA will at some time match the market needs, if you were ambitious enough at the UTOPIA definition stage. A warning: do not let the people write long specs corresponding to a more extensive software writing. This is another way to look at the DIVIDE AND CONQUERE Dijkstra statement, nothing else, from a perspective that has been suggested by technology evolution.

It is now important to underline that the availability of technologies for making applications to communicate among them makes realistic prototyping an extremely productive activity. Current efforts to develop those technologies make realistic prototyping an activity to be considered carefully with respect to the dimension of the project. Extremely large projects will be fully feasible (let me avoid discussing perfomance problems or platform resources that have to treated in another context) using realistic prototyping if interapplications communication will be available. If not, care must be exercized regarding the project dimension in relation to the high cost of redoing or doing.

INTERACTING WITH ARTIFICIAL REALITIES

Roberto Polillo
Dipartimento di Scienze della Informazione
Università degli Studi di Milano
and Etnoteam S.p.A. - Milano

"The screen is a window through which one sees a virtual world. The challenge is to make that worl look real, act real, sound real, feel real"

I.Sutherland, 1965

RIASSUNTO

L'interazione con le sofisticate 'realtà artificiali', realizzabili con le workstation di oggi pone nuovi e stimolanti problemi al progettista di interfacce utente. Nuovi paradigmi di interazione possono essere pensati e sperimentati. Sebbene questa disciplina sia ancora agli inizi, numerosi suggerimenti possono essere tratti dai computer games, dai sistemi ipermediali e dai prototipi sviluppati in molti laboratori di ricerca. La presente nota contiene una prima discussione di questi argomenti.

ABSTRACT

Interaction with "artificial realities" implementable on today workstations offers new challenges to user interface designers. New paradigms of man-machine interaction can be devised. The discipline is at its beginning, but many hints can be found in computer games, in hypermedia systems, and in prototypes being developed at many research sites. This note contains a first discussion on these issues.

1. INTRODUCTION

The interaction with "artificial realities" which can be implemented on today workstations offers new challenges to user interface designers.

New paradigms of man-machine interaction can be devised: some are based on traditional practices, while others are suggested by new *interaction devices* , and are still to be fully explored and evaluated.

The discipline is at its beginning, and systematic literature on the topic is practically non-existent. Nevertheless, many hints can be found in *computer games*, in *hypermedia systems*, and in prototypes being developed at many research sites.

This note contains a first discussion on these issues, with the help of some examples from commercial software products (mainly drawn from the computer games area).

A short italian version of this paper was presented at the Workshop "Conoscenza per Immagini", CNR, Rome, 27-28 November 1989 [39].

2. ARTIFICIAL AND VIRTUAL REALITIES

2.1 Artificial reality

"Artificial reality" means *a world inside a computer, with which humans can interact.*
We can implement many different kinds of worlds

inside a computer: purely mathematical worlds, worlds that mirror the physical nature in which we live, or even *wonderlands* with their own peculiar laws of physics or no laws at all (i.e., chaos).

With today's workstations processing power, and the achievements of computer graphics, we are able to visualize on computer screens complex artificial worlds with impressive visual details and appeal. In addition, we can make these visualizations change quickly and continuously, as we change our point of view on the scene.

These worlds can be explored and manipulated thorough suitable interaction devices. We can *travel* in them, *change* them, *interact* with them, and get accustomed with their *laws of nature,* which do not necessarily reflect the laws of our ("real") nature:

"The computer is a mirror in which reality comes to life again, out of space and time: the artificial reality" [12].

Traditional examples are professional *flight simulators,* and many kinds of *computer games,* ranging from flight and drive simulators to many *arcade games,* up to the celebrated *Dragon's Lair*, an *interactive cartoon* implemented some years ago on Laservision videodiscs.

Fig.1 shows an example of screen image from the *CT5-A* system, a commercially available top-level simulator by Evans & Sutherland of the early eighties; fig.11 shows the well-known Chessna *Flight simulator* from Microsoft [30], which inspired a number of fortunate home computer products (e.g., the more recent *Falcon,* a F-16 Fighter simulation by Spectrum HoloByte, [42]).

The above examples may be misleading, as they may suggest that an artificial reality, by definition, reflects the real, natural world in which we live (even if in an oversimplified way, as in many computer games).

This is not necessarily true. We may think, indeed, of artificial worlds completely different from the one to which we are accustomed: worlds of mathematical entities, worlds that represent the physical reality on a different *scale,* macroscopic or microscopic, worlds of fantasy as in many *shoot'em up* arcade games or in Stephen Lisberger's film *Tron,* or even worlds made up of documents and information, in which we *navigate,* as in the metaphor used by current *hypermedia* systems (e.g., see [15] and [19]).

The reader is referred to [10] for examples drawn from the scientific modeling area.

2.2 Virtual reality

The terminology which is currently used is not uniform at all: in the literature, terms such as "artificial reality", "virtual reality", "synthetic worlds", "virtual worlds", ... are used more or less interchangeably.

In this paper, following the distinction suggested by G.Degli Antoni, we will use the term **virtual reality** (instead of "artificial reality") in all those cases in which

there is some kind of interaction between the artificial world and the real world, in such a way that by interacting with the artificial world it is possible to interact with the real world, and viceversa

In this way, for example, we can control the real by controlling a representation of it (and viceversa), as it happens in classical *process control systems, building automation systems,* and so on.

But we can imagine much more complex examples of virtual realities, in which the interaction between "the real" and "the artificial" is much tighter that in simple process control.

For example, in [12] G. Degli Antoni describes an imaginary application in the area of public services. A Public Office (e.g., a Ministry) is represented on the computer screen, with all its corridors, doors, clerks and archives. The user can enter and move in it, opening doors with the click of a mouse, consulting the appropriate archives and asking questions to the "artificial clerks" that he encounters. When the artificial clerk cannot satisfy the visitor needs, it *calls the real clerk at the real Ministry,* who in some way appears on the screen and speaks directly to *both of them.*

Note that, here, the interaction is threefold: between the

user and the artificial world, between the artificial world and the real world, and between the user and the real world.

The distinction between artificial and virtual reality is schematized in Fig.2.

3. ARTIFICIAL REALITY AND USER INTERFACE

To interact with artificial (or virtual) realities, we need a suitable **user interface**, through which we can **visualize**, **explore** and **change** such worlds.
Different **interaction paradigms** can be devised, depending on the **interaction devices** that we use, and possibly on the **metaphors** that inspire us when designing the user interface.

Studying and experiencing these possibilities constitutes a new challenging discipline, which is still at its beginning. The available literature is still very limited. Nevertheless, many hints can be found in different application areas, some already well consolidated: computer games, hypermedia systems, new interaction devices, simulators, and so on.

By examining these examples, we propose in the present paper a first classification of interaction paradigms

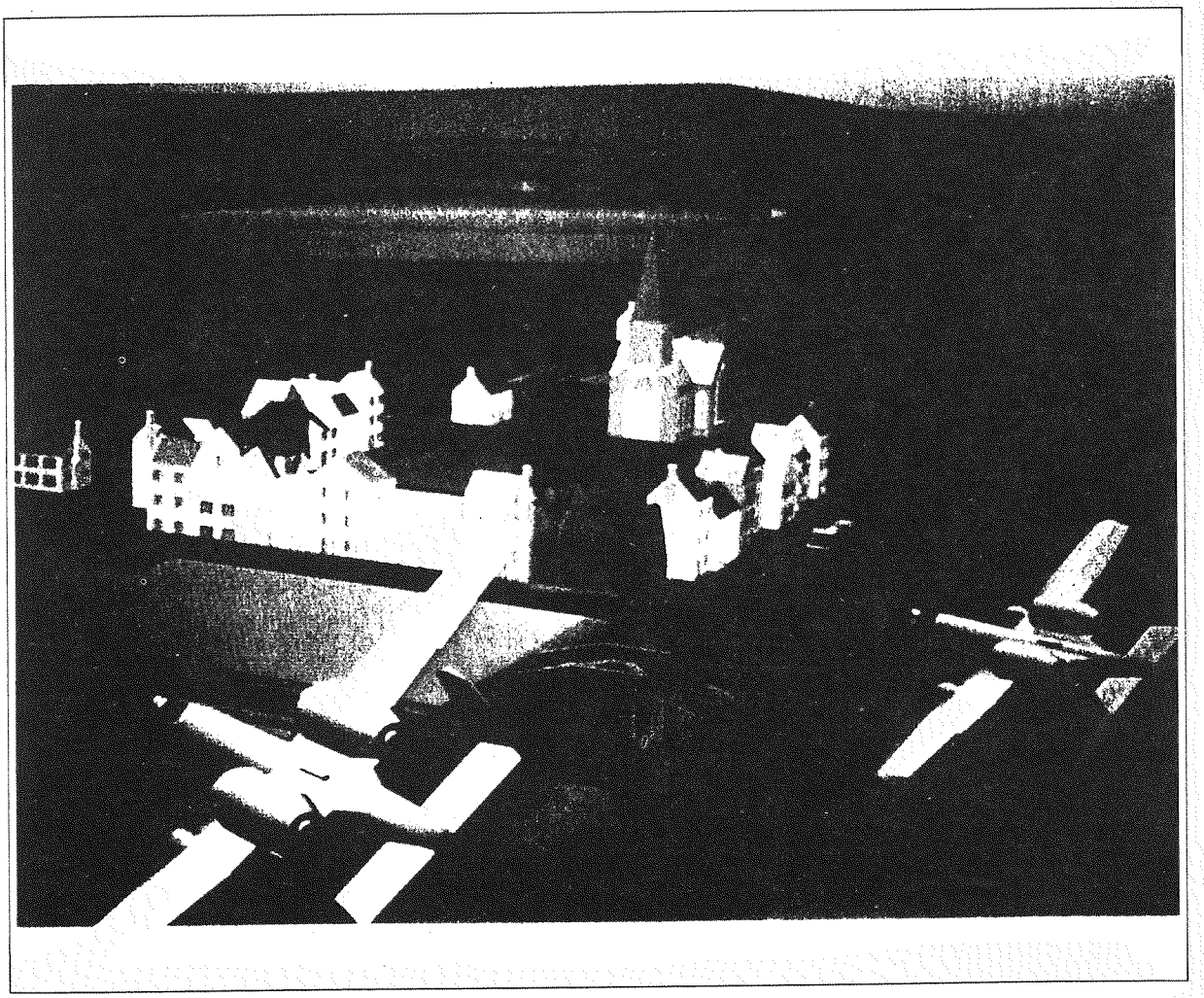


Figure 1: Evans & Sutherland CT5-A Simulator

with artificial realities in five classes:

- **command languages**
- **menu systems**
- **direct manipulation**
- **navigation**
- **virtual presence.**

These paradigms are summarized in Fig.3, and will be briefly examined in the following paragraphs.

4. COMMAND LANGUAGES

Command languages were the typical paradigm of man-machine interaction when the dialogue with a computer was performed by means of *teletypewriters*.

In artificial reality related applications, this is the typical paradigm used in classical *text adventure games* (from the early *Adventure* written by W. Crowther for the PDP-10 in 1975 to the best-selling Infocom games, beginning from *Zork I: The Great Underground Empire* by M. Blank and D. Lebling, see [29]).

In these systems (Fig.3a), the artificial reality (i.e., the world of the adventure) is not visible, but "imagined" by the player, who controls its own *alter ego* that lives and moves and fights (and possibly dies) in it.

Interaction is performed by means of a **pseudo-natural language,** usually with a very simple English-like syntactic structure (but the lexical structure - i.e., the *dictionary* of the game - is often fairly rich). In many

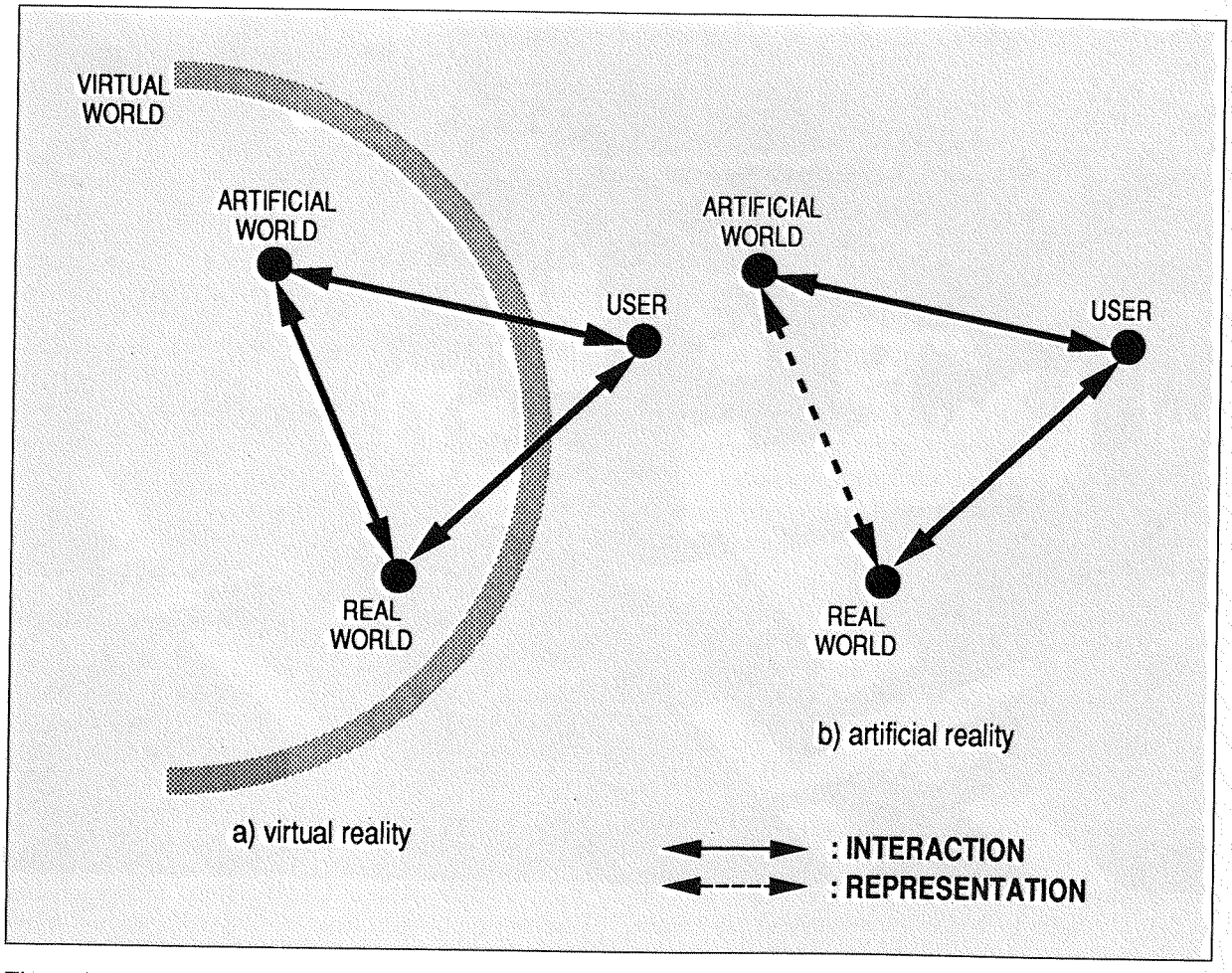


Figure 2: Artificial vs virtual worlds

cases, the parser will only recognize a number of *keywords*, as in the classical example of Elisa, a "program pretending to be a doctor", written by Weizenbaum more than 25 years ago [47].

As an example of dialogue, the following is one of the possible beginnings of *Hitchhiker's Guide to the Galaxy*, another well-known Infocom game, [24] (the text typed by the player is capitalized):

>SWITCH THE LIGHT ON
Good start to the day. Pity it's going to be the worst of your life. The light is now on.
Bedroom, in the bed.
The bedroom is a mess.
It is a small bedroom with a faded carpet and old wallpaper. There is a washbasin, a chair with a tatty dressing gown slung over it, and a window with the curtain drawn. Near the exit leading south is a phone.
There is a flathead screwdriver here. (outside the bed)
There is a tootbrush here. (outside the bed)

>GET OUT
(bed)
Very difficult, but you manage it. The room is still spinning. It dips and sways a little.

>GO TO THE DOOR
It's here!

>OPEN THE DOOR
It is already open.

>GO THROUGH THE DOOR
Did you have any particular direction in mind?

>NORTH
You can't go that way

>SOUTH
You miss the doorway by a good eighteen inches. The wall jostles you rather rudely.
.....

The main problem with this kind of user interface (which is evident in the above example) is that the player is continuously forced to solve linguistic puzzles ("*How am I supposed to phrase the command that I*

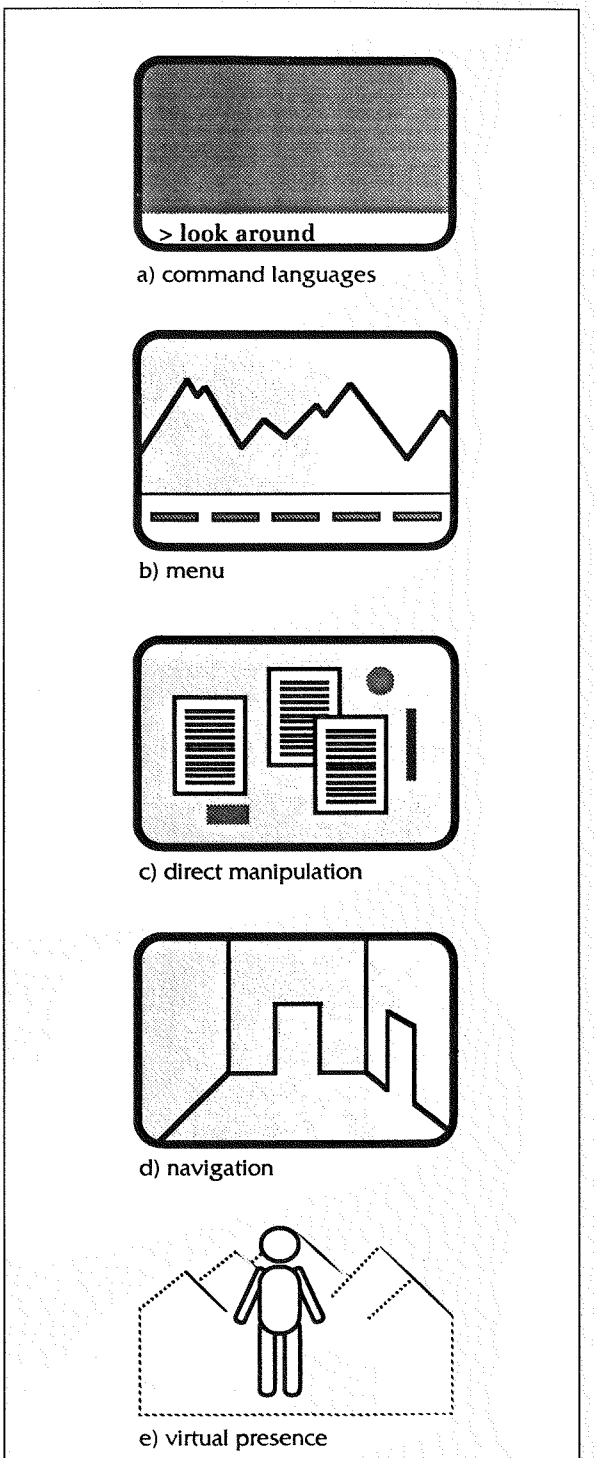


Figure 3: The five main interaction paradigms

want to issue?"), which have nothing to do with the domain of the adventure.

For this reason, once Ted Nelson called these systems "guess my command games".

On one hand, this can be considered a challenge to the player intellectual abilities (and might have played an important role in the commercial success of these games). On the other hand, this kind of dialogue violates one of the fundamental principles of good user interface design, i.e. the *principle of transparency*, which is well expressed in the following quotation by Winograd and Flores [48]:

"In working with people, we establish domains of conversation in which our common pre-understanding lets us communicate with a minimum of words and conscious effort. We become explicitly aware of the structure of conversation only when there is some kind of breakdown calling for corrective action. If machines could understand in the same way people do, interactions with computers would be equally transparent. (....) A bad design forces the user to deal with complexities that belong to the wrong domain."

5. MENUS

Menus are one of the simplest and most widespread paradigms of man-machine communication, from the

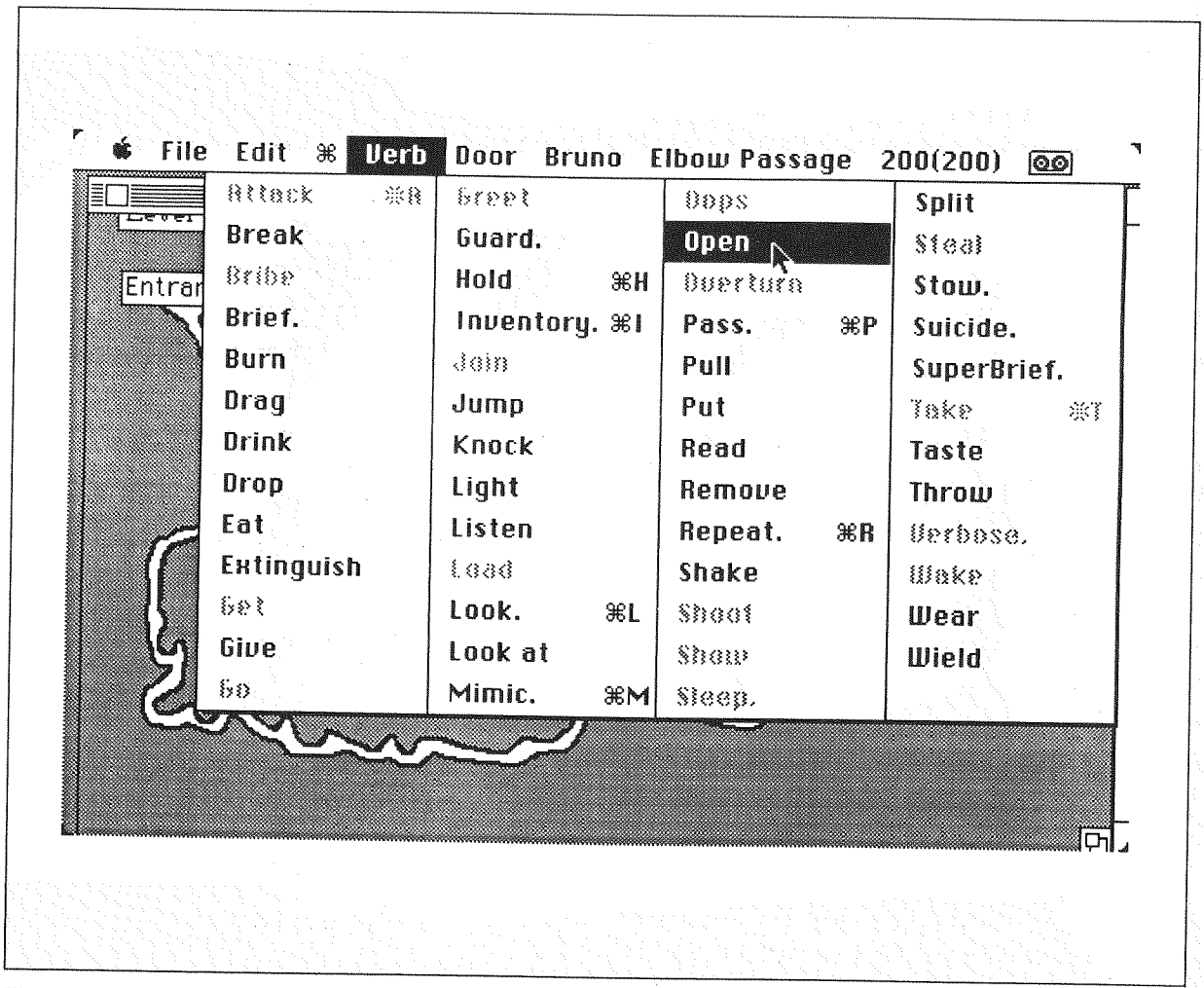


Figure 4a: Quarterstaff: a pull-down menu

times when the standard interaction device was the (nonprogrammable) *alphanumeric terminal* or *VDU* .

Typical applications used **full screen menus**: the VDU screen displayed only a list of alternative actions, and the user was asked to select one. The selection typically resulted in the display of a *lower level menu*, or of a *form* for entering and querying data. (This paradigm of interaction has been recently standardized by IBM, in its Common User Access *Entry Model* interface, [22]).

The old-fashioned menu paradigm had a new evolution with the diffusion of personal computers, when the *local* processing capabilities of the machine allowed a faster interaction between the user and the application. The most of the screen space was then reserved to the application data (for example, collected in a spreadsheet), while the menus were confined to a small screen area, typically a single line (**single line menus**).

In single line menu systems, two main paradigms emerged:

- the **menu bar paradigm**, popularized by the *Lotus 1-2-3* spreadsheet;
- the **pull-down menus paradigm**, popularized by the Apple Macintosh, and thereafter adopted in a large number of PC applications, and even on nonprogram-

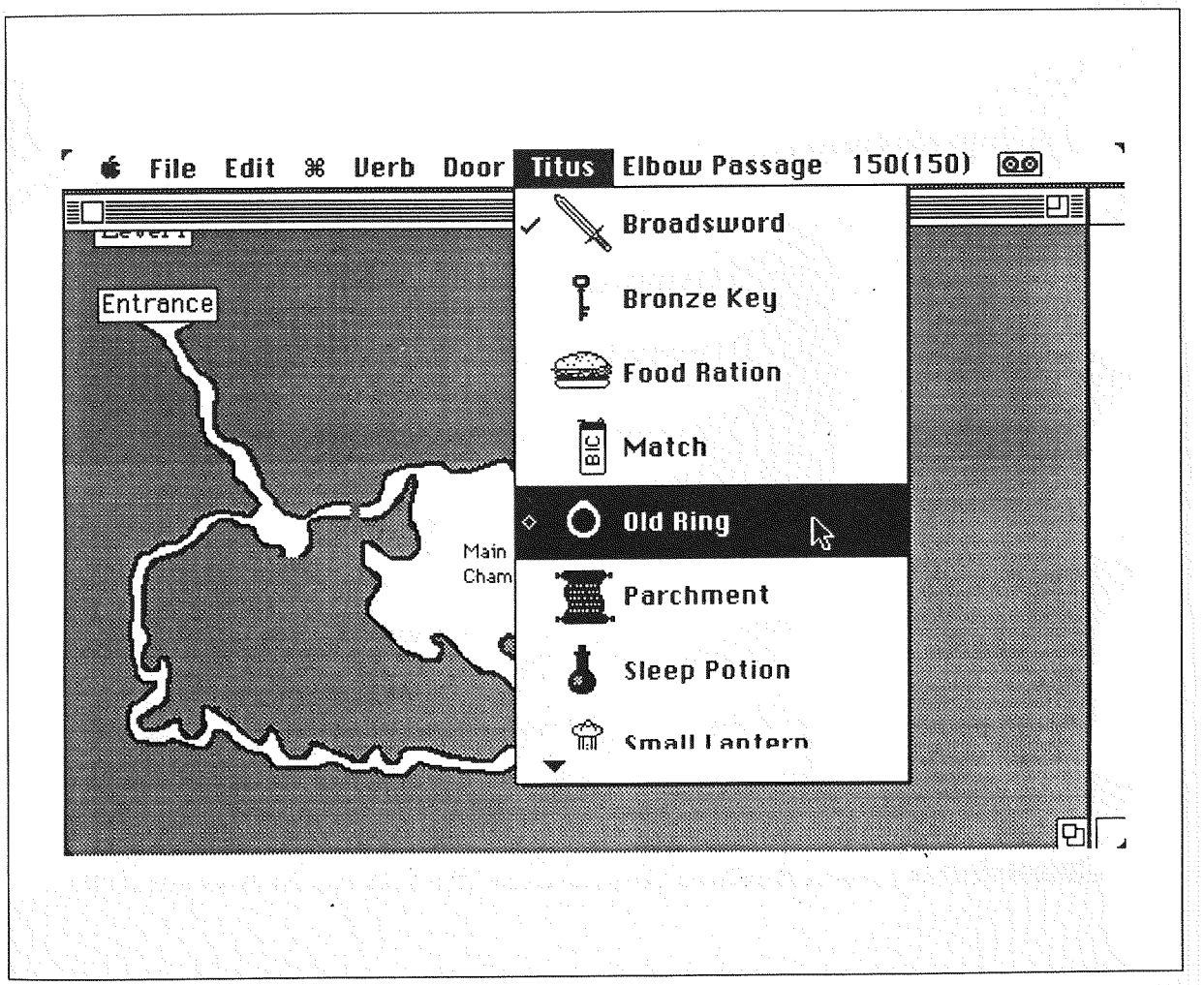


Figure 4b: Quarterstaff: map and iconic menu

mable alphanumeric terminals (as for example in the *Text Subset of the Graphical Model* of IBM Common User Access standard, [22]).

In artificial-reality related applications, both kinds of single line menus have been used, from the mid eighties, in a number of adventure games exploiting the graphical capabilities of the home computers.

In the first games of this kind, the available commands were selected from a menu bar displayed (for example) at the bottom of the screen. In the rest of the screen, the world of the adventure was visualized (even if still in a rudimentary way) by displaying a *static picture* of the current scene (Fig.3b).

In other cases the available commands are selected from one or more pull-down menus. This is, for example, the case of *Quarterstaff*, recently published by Infocom for the Macintosh, [25], were a few menus allow one to specify the desired command. Here the graphical capabilities of the Macintosh are used to display a *map of the world,* showing the current location of the characters, or the portrait of a new character, or even *pictograms* to make the menu items clearer (Fig.4a and 4b)

6. DIRECT MANIPULATION

According to Shneiderman, the central ideas of **direct**

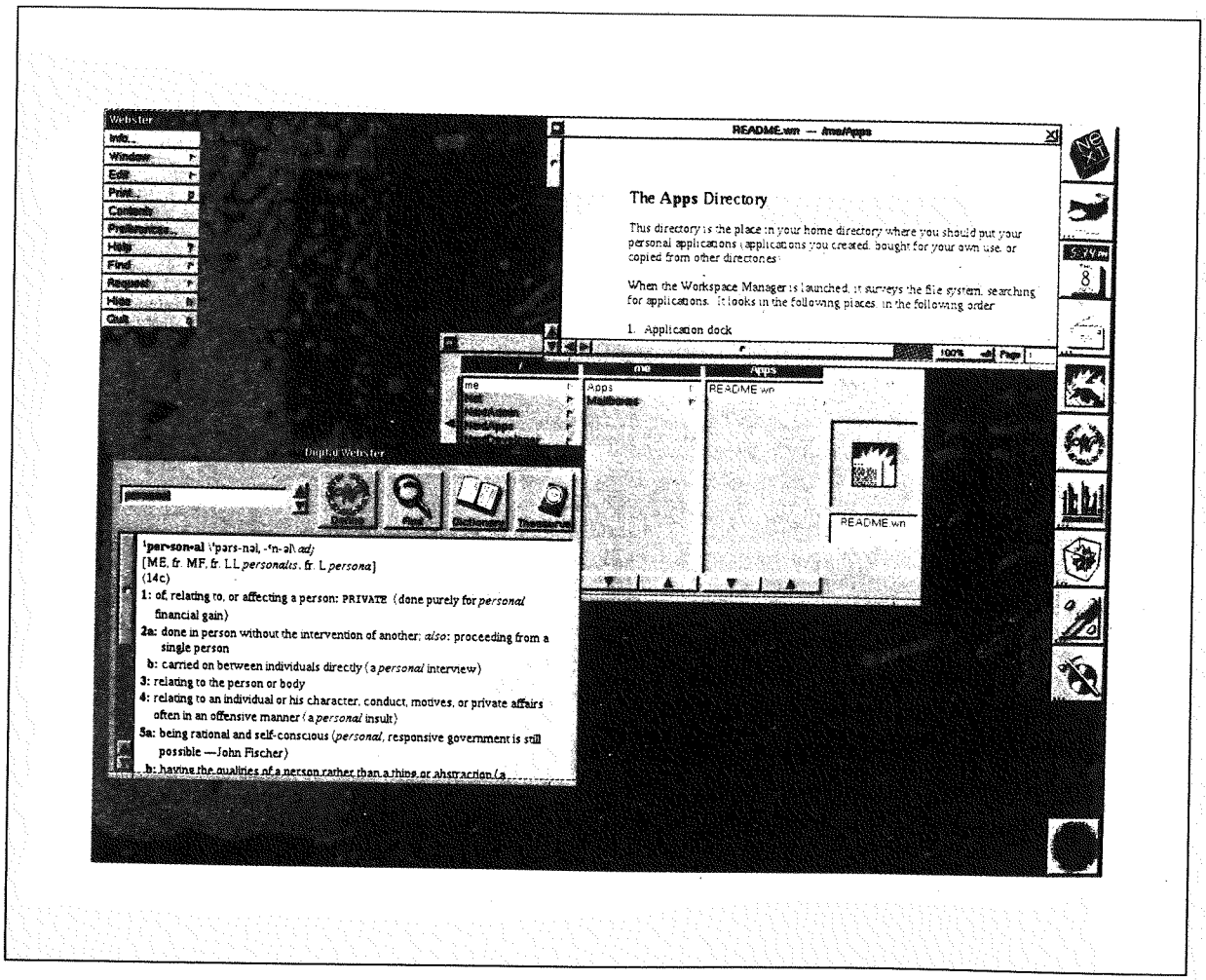


Figure 5: The NeXT Computer workspace

manipulation interfaces *"seems to be the visibility of the object of interest; rapid, reversible, incremental actions; and replacement of complex command language syntax by direct manipulation of the object of interest"* [40].

This paradigm probably has its roots in early full-screen editors, in which the text at the cursor position was easily manipulated with the help of a variety of **function-keys commands** (EMACS, vi, Wordstar,...). Today, the typical examples of direct manipulation can be found in the **mouse** based interfaces of current workstations or personal computers (Fig.3c).

6.1 "Flatlands"

The best known examples are the systems based on the so-called **desktop metaphor**, from the Xerox Star [41] to the Apple Macintosh *Finder* [5], the OS/2 *Presentation Manager* [23], the Sun Microsystems *OpenLook* ([43], the OSF *Motif* [37], up to the sophisticated NeXT interface [36].

In these systems, with the growing graphical capabilities of current workstations, the original flat, 2D representations of the desktop objects progressively get a kind of *bass-relief look*. This is particularly remarka-

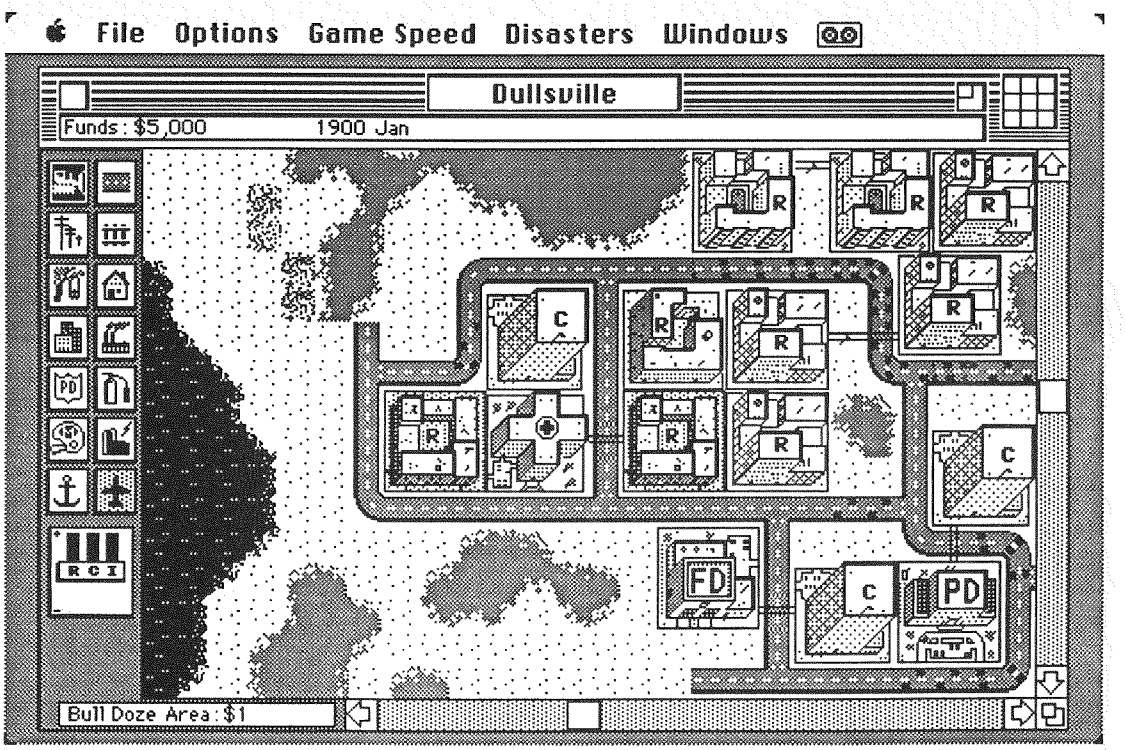


Figure 6: SimCity

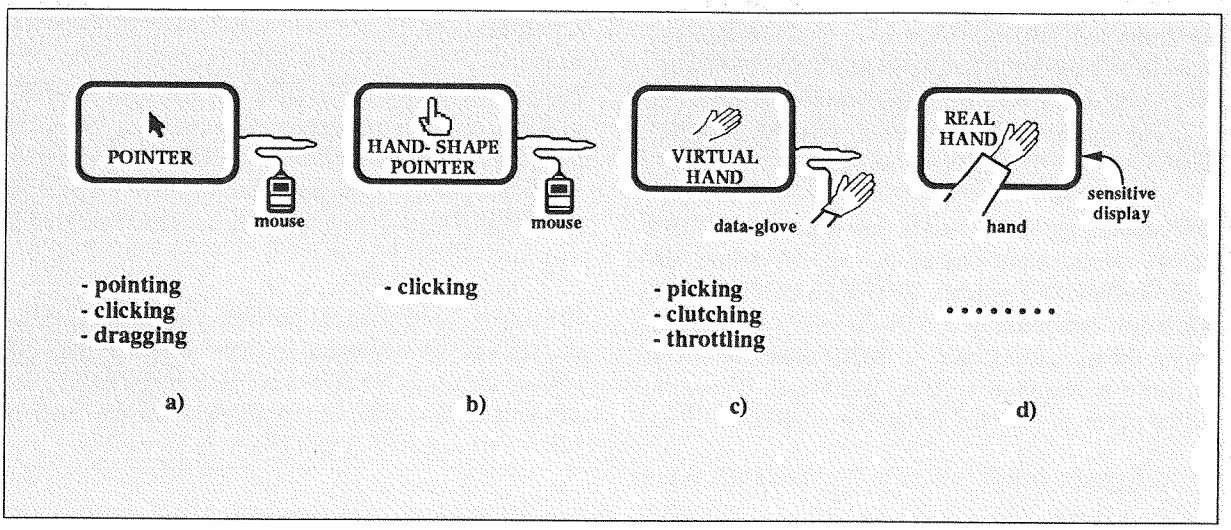


Figure 7: From the "cursor metaphor" to the "hand metaphor"

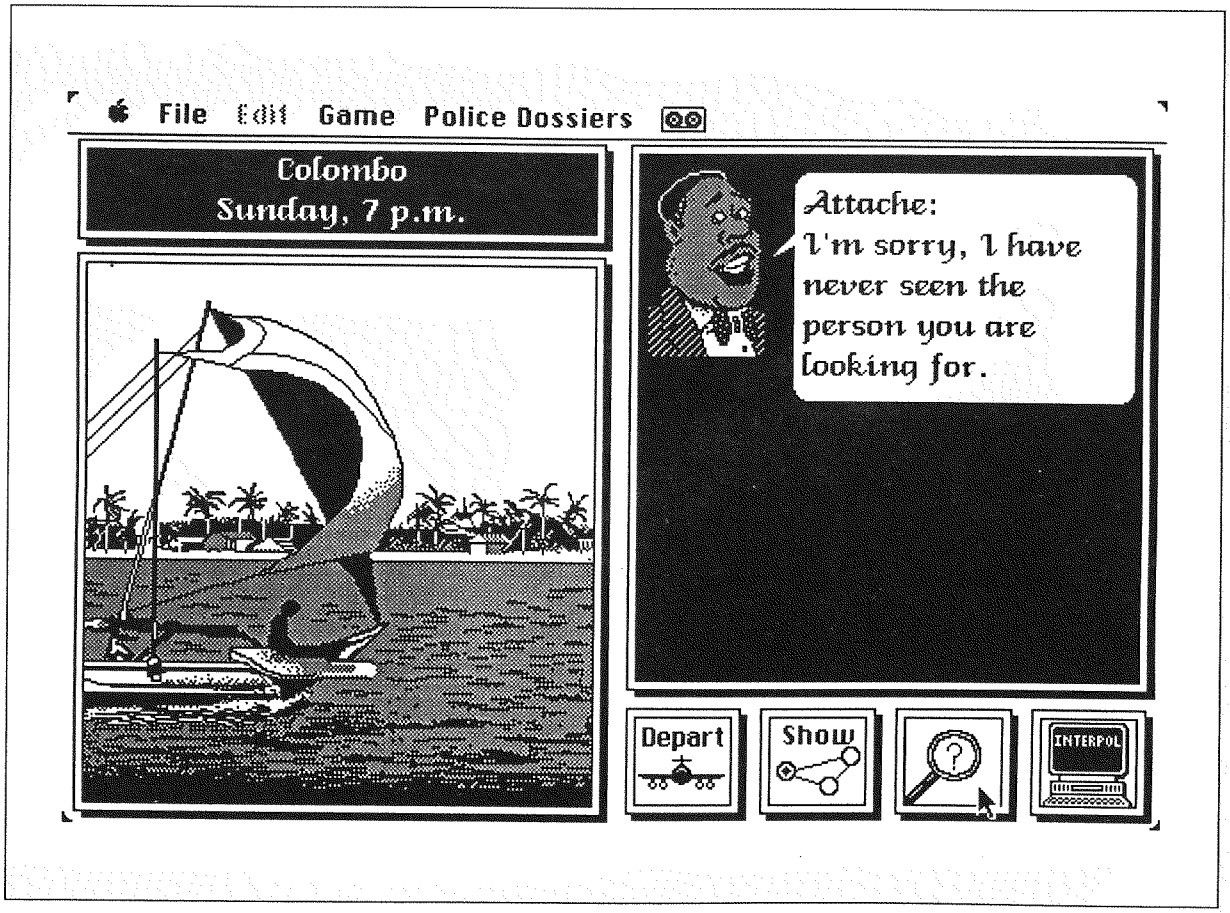


Figure 8: Where in the world is Carmen Sandiego

ble in the NeXT interface, for which we could more properly speak of a **control-panel metaphor** (Fig. 5).

All the above systems were primarily designed for office use: therefore, the artificial worlds that they represent on the screen are made essentially of documents, folders, trashcans, and so on.

An example of a different kind is shown in Fig.6. The example is taken from *SimCity*, a recent Macintosh game published by Broderbund [9], which allows one to create, change and simulate the behaviour of towns. The picture shows the top-view of a town during the simulation. The user can manipulate the city in many ways (e.g., building roads or destroying bridges) with a number of tools which can be selected, using the mouse, from the *tooltray* shown on the left.

Despite the bass-relief appearance of some of them, all these examples represent essentially 2D worlds, and therefore we may call them "Flatlands", as in E. A.Abbott and C. H.Hinton well-known fantasies [1] and [20].

With them we interact directly *with our hand*, represented by the *mouse pointer*, and through the use of an elementary **gesture language**: simple, standardized gestures denote the actions that we want to perform (e.g., *clicking* means "select this one", *double-clicking*

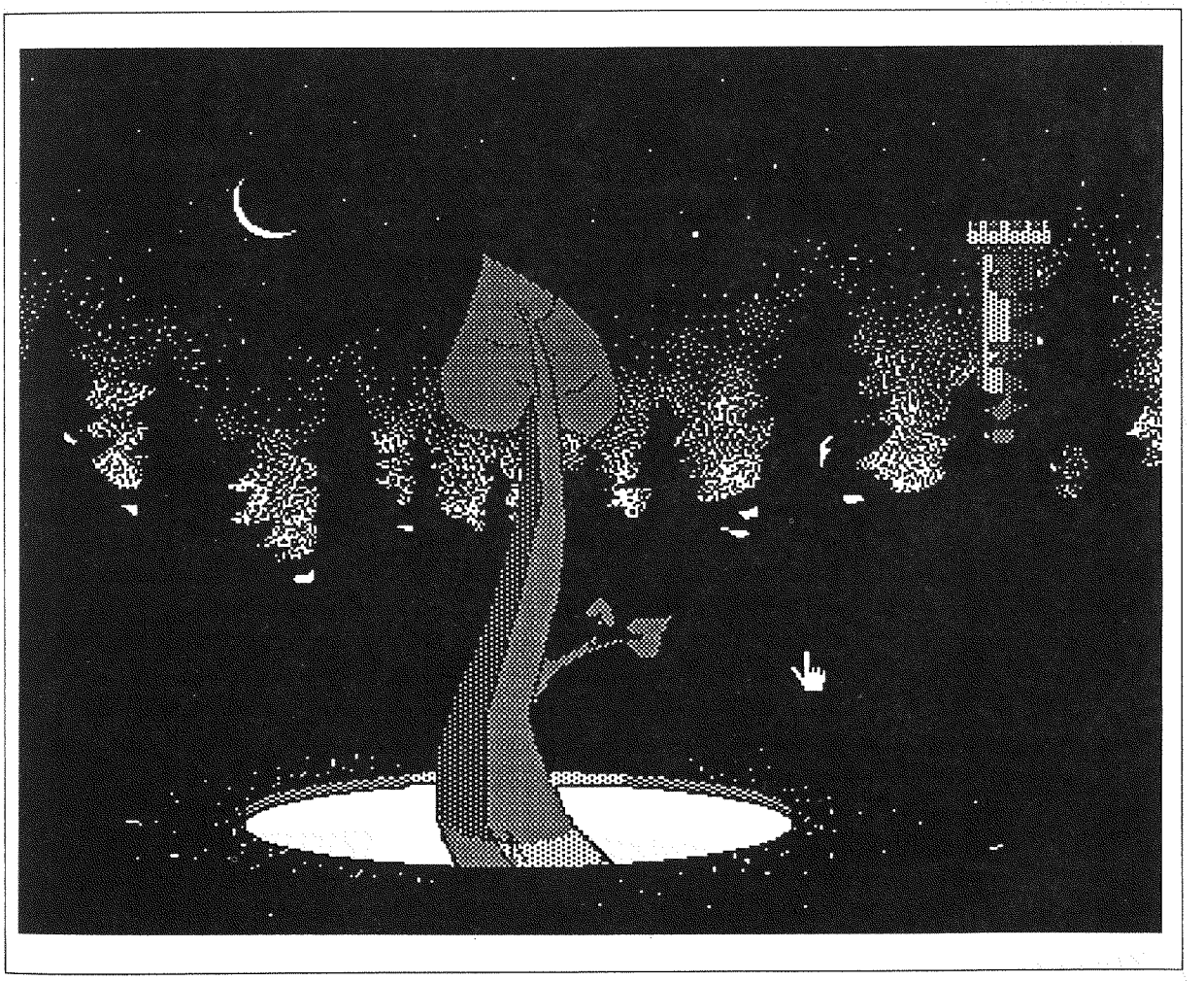


Figure 9: The Manhole

means "open this one", and so on).

6.2 From the cursor metaphor to the hand metaphor

Starting from the traditional, twenty-five years old mouse (ENGL67), which may have one, two or three buttons, new *direct manipulation devices* can be designed. Examples are the **touch sensitive screen,** the **trackball,** the **data-glove** [16], the **force and position sensitive screen** [35].

It is very interesting to observe that direct manipulation systems show a slow, progressive evolution from the *cursor metaphor* to the *hand metaphor* (see Fig.7).

6.3 Mouses and hand-shaped pointers

A first step can be seen in *Hypercard* [6], where the Macintosh mouse-controlled arrow pointer (Fig.7a) trasforms into an "hand-shaped pointer" (Fig. 7b), while the gesture language is further simplified: the small Hypercard hand can only *point* and *click* (other actions, such as *dragging* or *double-clicking* are eliminated).

With this simplification, user interfaces can be based on a single, elementary but general idea: the **button metaphor**. Buttons may be either visible or invisible, may have many different shapes and labels, may visually

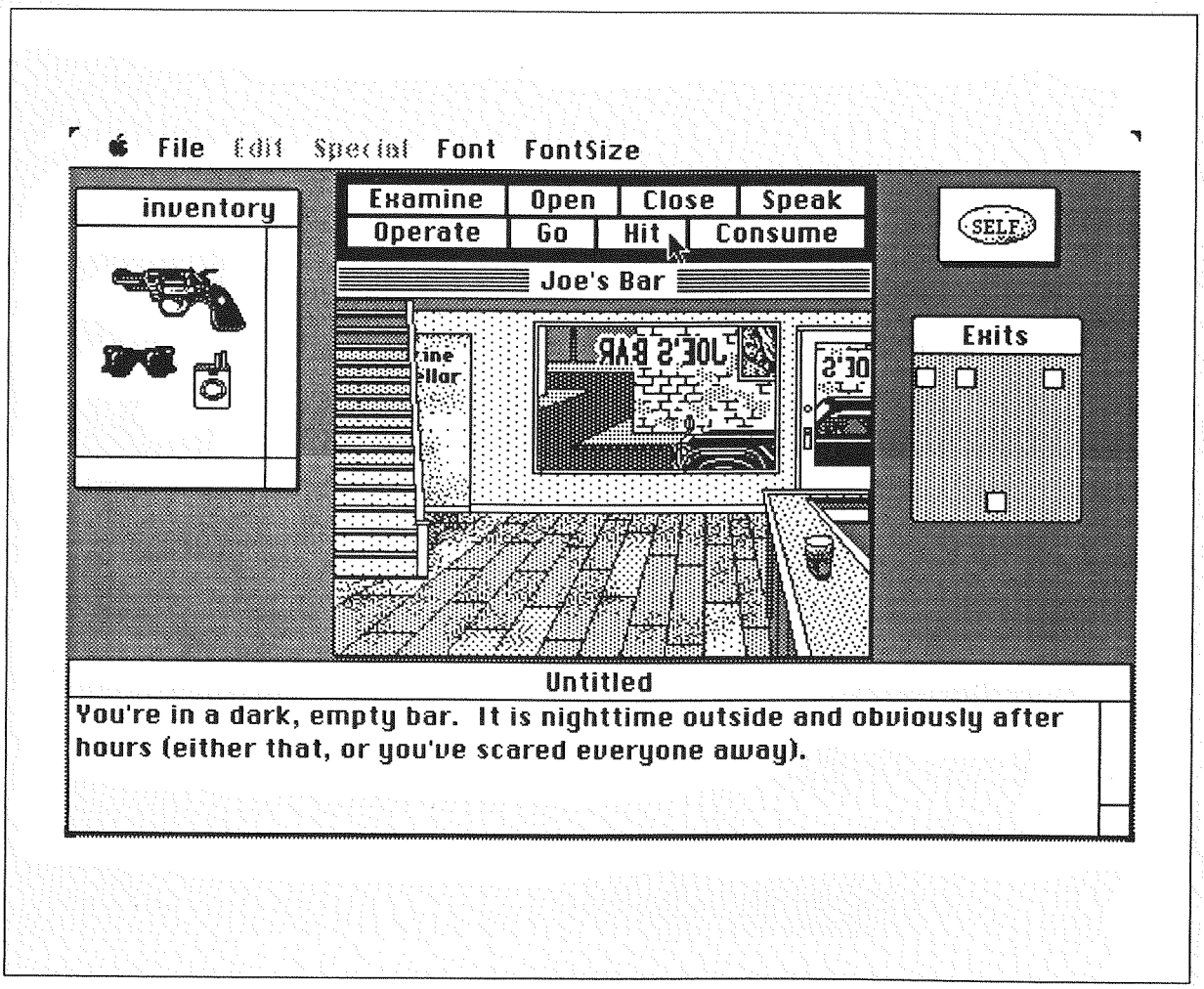


Figure 10: Deja Vu

respond in different ways to user clicks, and may be associated to *scripts* which are executed when the button is pressed.

6.4 Data-gloves and virtual hands

A very interesting device, which is already commercially available but still has to be fully exploited, is the *data-glove*. This is a thin glove with optical fiber sensors that detect how the hand is bending, developed by VPL Reasearch Inc., Palo Alto, California (e.g. [16], [45], [17]).

A number of experiences are being carried on using this innovative device. For example, in an experimental user interface developed at Bell Labs [46], the traditional mouse pointer changes in a small "virtual hand", controlled by the user "real hand" wearing the data-glove and moving on the "real" desktop.

The actions to be performed on the (flat) artificial objects represented on the screen are expressed with a simple gesture language, including gestures such as "picking", "clutching", "throttling", and so on (Fig.7c).

6.5 Sensitive screens and real hands

In Margaret Minsky's research at MIT Media Lab [35], also the virtual hand is eliminated, and replaced by a "real" hand: an experimental type of sensitive screen can sense single finger gestures in terms of position, pressure and shear force (Fig.7d).

Examples of applications are *fingerpainting* on the screen, moving virtual buttons around with the finger pressure (or even "flicking" them), and so on:

"We want to create worlds within the computer that can be manipulated in a concrete natural way using gesture as the mode of interaction. The effect is intended to have a quality of "telepresence" in the sense that, to the user, the distinction between real and simulated physical objects displayed on the screen can be blurred by letting the user touch, poke, and move the objects around with finger motions" [35]

7. THREE EXAMPLES

Even if we limit ourselves to the mouse as interaction

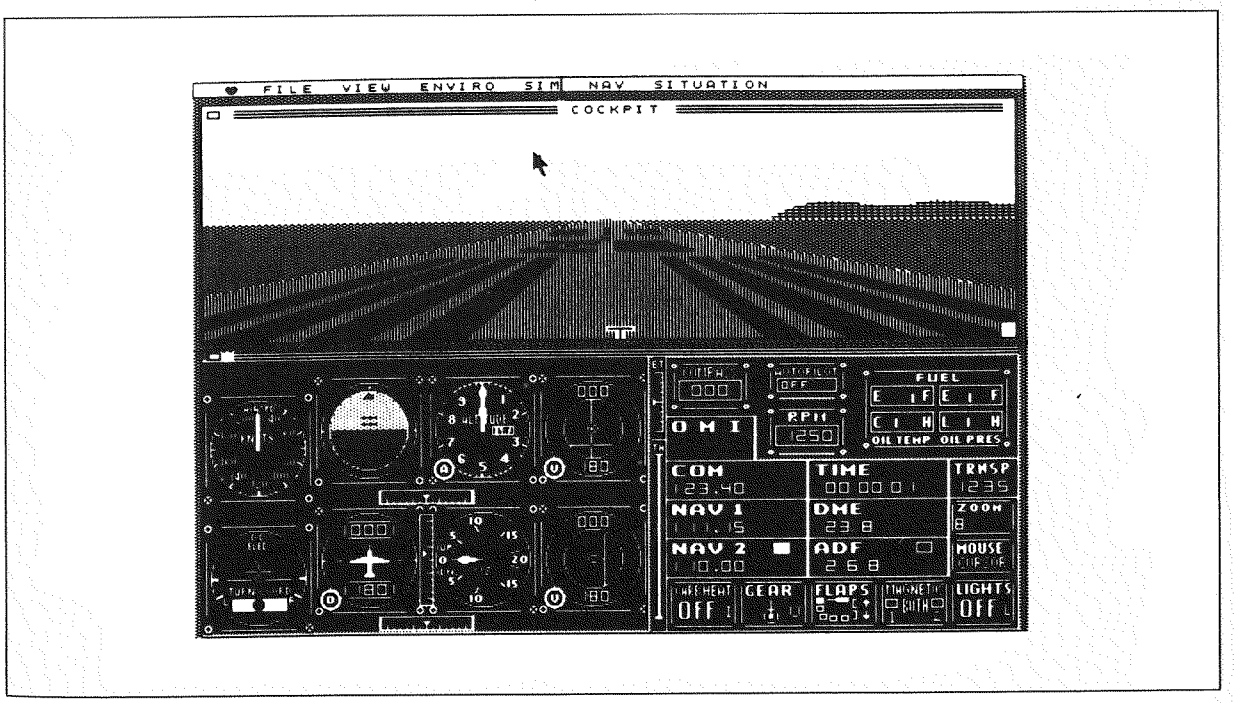


Fig.11: Flight Simulator

device, the possibilities of direct manipulation interaction with artificial worlds are significantly richer than those available in command languages or menu systems, and many different techniques can be devised.

Here below, we shall briefly describe three typical examples, of increasing complexity, of traveling (and possibly acting) in an artificial world, visualized through a number of static pictures. All of them are recently published (and best-selling) Macintosh games.

The simplest example is the educational game *Where in the world is Carmen Sandiego?* from Broderbund [85]. Here the player must travel to different places all over the world to find and arrest Carmen Sandiego or one of her criminal fellows. To fly somewhere, he simply clicks the desired place on a *world map*, and gets a picture of the destination site on the screen. Here, a few, simple actions can be performed, by pressing one of a small number of *buttons* (Fig.8).

The second example is *The Manhole*, an **interactive storybook** for children developed with Hypercard and published by Activision. Here the artificial reality is visualized by means of a number of static pictures or "snapshots" (i.e. Hypercard *cards),* on which many **invisible buttons** are layed out.

By pressing these buttons, the scene on the screen changes, in a way which is often surprising. (Fig.9, [3]). In this case, the artificial reality is made of a stack of **sensitive pictures**, which are linked one another in a complex, large **hypergraph**.
Cosmic Osmo, another recent interactive storybook developed by the same authors, uses the same basic ideas and technology, with the addition of simple

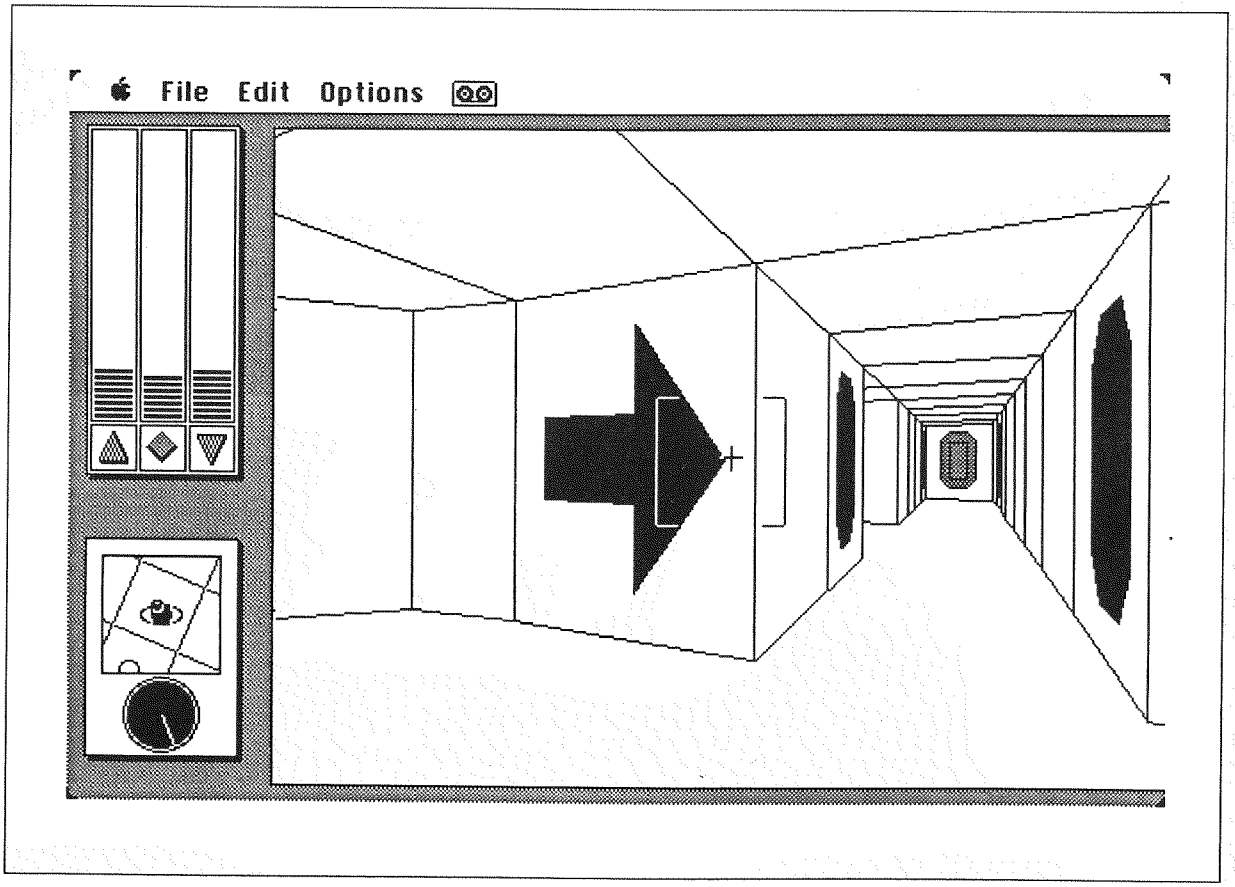


Figure 12: The Colony

animation: some cards display objects (or characters) which animate, for example when clicked with the mouse [3].

A third, more sophisticated example of traveling and acting in an artificial world can be found in Mindscape's adventure games *Deja Vu* [32], *Uninvited* [33] and *Shadowgate* [34].

Here (Fig.10), we find a menu of action buttons, which can be pressed with the mouse. An *action/object paradigm* is used: to open a door, for example, one must click the "OPEN" button, and then the door in the picture (or, simply, double-click the door).

Pictures of the world are static, but most objects are moveable, and can be *dragged* around the screen with the mouse, for example to move them to the "Inventory window" (in which all the objects carried by the player are shown).

8. NAVIGATION

This is the typical paradigm which comes to mind when speaking of artificial reality. According to this paradigm, the computer screen visualizes a *tridimensional* world, in which we are *immersed,* and in which we *navigate with continuous movements* using an appropriate *driving system* (**virtual travel**).

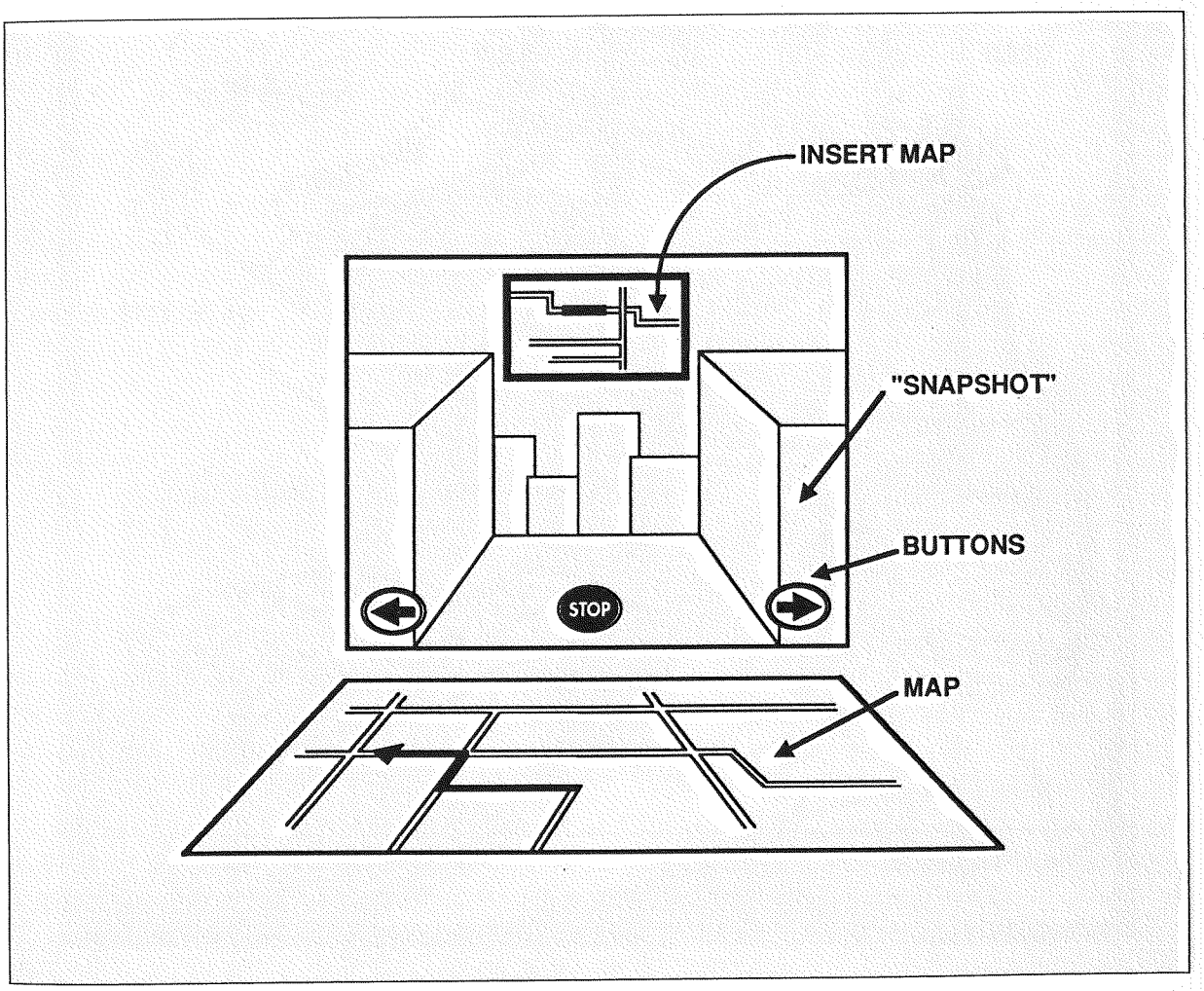


Figure 13: The "Movie Map" of Aspen, Colorado

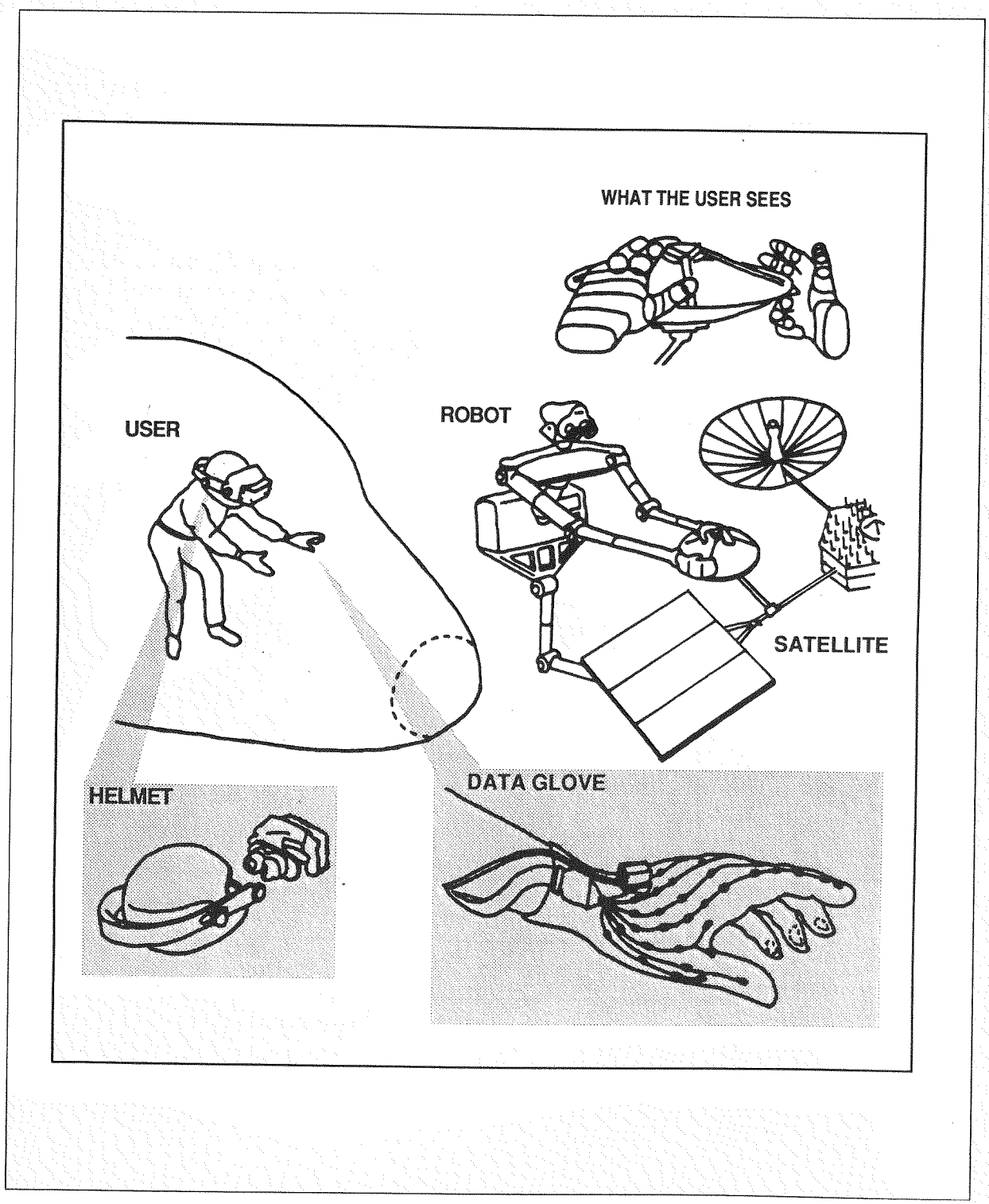


Figure 14: Telepresence

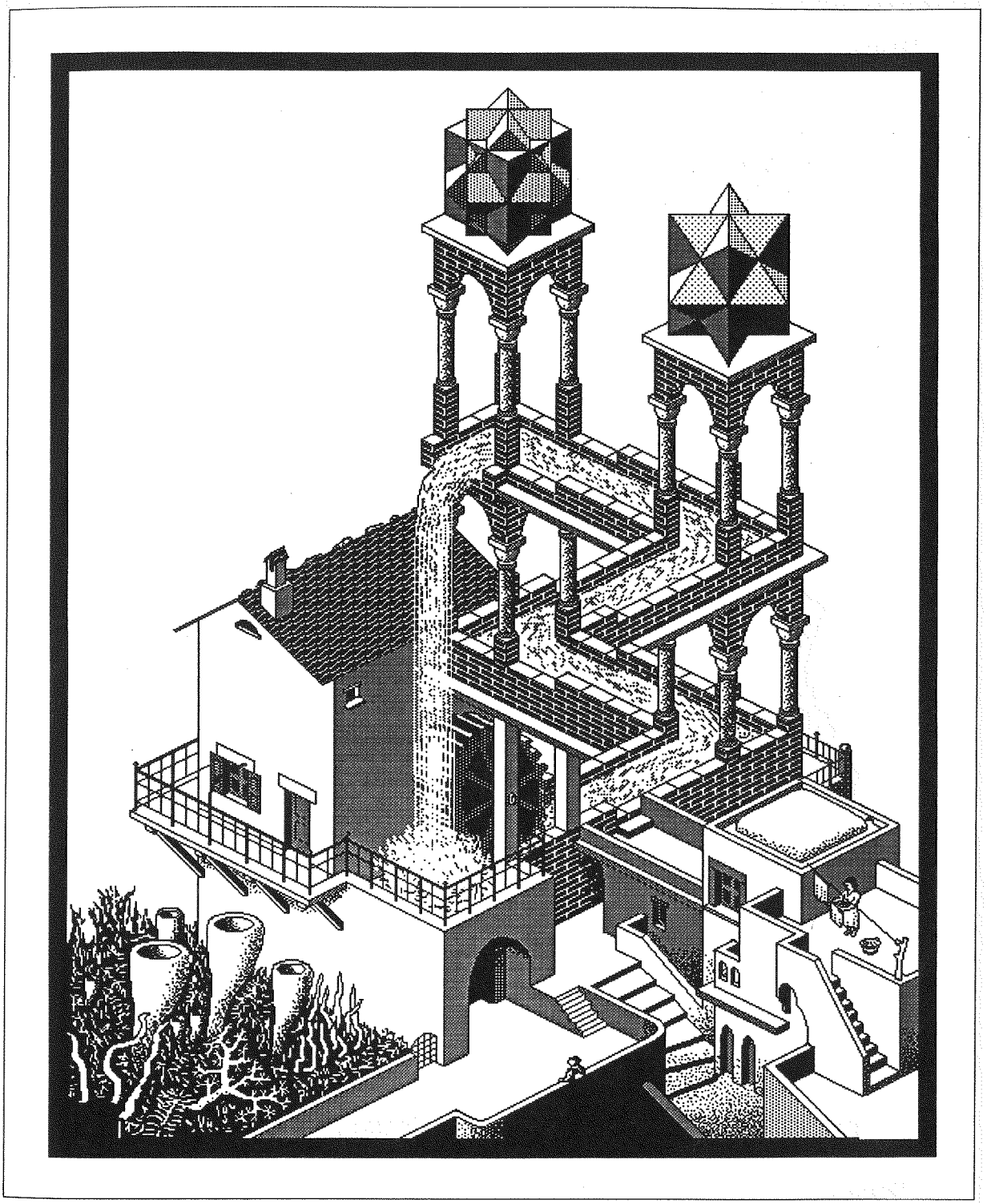


Figure 15: M.C.Escher, "Perpetual Motion", 1961

8.1 Driving systems

We can use various devices to drive the virtual travel in the artificial world: a *joystick*, a *bat* (a sort of mouse which can be moved in the space and not only on a plane) or even a traditional *mouse*.

Various metaphors may suggest the *driving paradigm:*

- in the *helicopter metaphor*, the "user eye" moves, rotates and stands freely in a tridimensional space, as if the user were driving and helicopter;
- in the *plane metaphor*, the user "flights" as it were driving a plane. Usually, he looks ahead, but can, at will, "turn the head" and look left or right (or even back). A typical example is the Microsoft *Flight Simulator* ([30], Fig.11);

- in the *car metaphor*, the user moves on the xy-plane, as it were driving a car, and can, at will, look left and right (and perhaps back);

- in the *shopping-cart metaphor*, the user moves on the xy-plane, but can only *look-ahead*, and not *look-aside*..

An interesting case is the adventure game *The Colony*, published by Mindscape ([31], Fig.12), which has a notably simple driving system.

Here the navigation is performed using a *space helmet metaphor*: the player sees the tridimensional world through an imaginary helmet, in which a cross-hair indicates the center of the scene (see Fig.12). The player moves around by moving the mouse pointer in the cartesian plane of which the cross-air is the origin: the y-axis is the forward and backward velocity control, while the x-axis is the left and right angular velocity control.

In simpler terms: move the pointer above the cross-air and you go forward, move it under and you go backward, move it to the left and you turn left, move it to the right and you turn right. If the pointer is near the cross-air, you move slowly; if the pointer is far away from the cross-air, you move fast.

In many of these systems, user orientation is sustained by a bidimensional *map* of the artificial world, or similar devices (in Fig.12, for example, an *overhead view* and a *compass* can be seen in the lower left corner).

8.2 "Snapshots" systems

In some (earlier) cases, the travel within the artificial world is visualized by a sequence of statical, "contiguous" snapshots as, for example, in the classical *Movie Map* prototype, developed some years ago at the MIT Architecture Machine Group [28].

In this system, movie footage was taken of travel down all the streets of the town of Aspen, Colorado, one frame per every 9 feet of travel, by a set of truck-mounted cameras. This footage was then transferred to videodisk. With special supporting programs to control image presentation from videodisk, the user can travel about Aspen at street level. Left and right turns at any intersection can be made by touching left and right arrows graphically overlaid on the touch sensitive screen of the user's color TV monitor (see Fig.13).

A tiny insert map at the top-middle of the screen shows the user's route with respect to Aspen two main streets. In addition, a second, horizontally mounted monitor provides a top-down view of the streets grids of Aspen like a conventional map. Again, the user's route is indicated in red on this monitor [7].

9. VIRTUAL PRESENCE

The final paradigm that we shall consider is still advanced research area; we shall call it *virtual-presence*.

The best-known example is a research system developed at the Aerospace Human Factors Research Division at NASA's Ames Research Center in Mountain View, California ([16], see also [45] and [17]).

The system is called *Virtual Interactive Environment Workstation (VIEWS)*, and uses head-tracked and head-mounted displays, gesture tracking, and voice input and output in an integrated way.
The user wears a special helmet that contains two small display screens, one for each eye, so that the image appears to be three-dimensional. The helmet prevents

anything except the image from being seen, immersing the user in the simulated scene.

A sensor mounted on the helmet keeps track of the position and orientation of the user's head. As the head turns, the computerized scene shifts accordingly. The system lets the user explore a 360-degree field of vision (that can be either a simulated or remotely sensed environment) and have *direct* interaction with its virtual components.

To interact with objects in the simulated world, the user wears a data-glove: so *he can grasp an move virtual objects with his (real) hand* (a separate sensor, similar to the one in the helmet, determines the hand's position in space; feedback to the user is provided by displaying a virtual hand in the simulated scene).

VIEWS also incorporates a commercially available sound system: with headphones, the user can hear sounds, either from within or outside his immediate field of view. In addition, a speech-recognition system lets the user issue vocal commands.

With this kind of apparatus, a human could, for example, command the arm of a robot located in a dangerous environment: as the user head moves in the helmet, TV cameras on the robot would transmit the robot view of the scene *(tele-robotics)*.

NASA envisions using this technology to allow an astronaut inside a space station to operate a robot making repairs outside (*telepresence*, Fig.14).

10. KINESTETICS, ARTIFICIAL REALITY, REALISM

In designing the human interface with artificial reality systems, it is necessary to understand and apply the *principles of kinestetics* which are most appropriate to the chosen paradigm.

The main problem, here, is to avoid the phenomenon of *user disorientation* , which may occur when we find ourselves in environments which are unfamiliar, or different, or too complex, or simply, for some reason, *disconcerting* to the human being.

The understanding of these principles is still very limited (for example, [10] lists a few "rules of thumb" regarding 3D interfaces). They will have to be systematized in a new discipline, which we may call **artificial kinestetics** *("the perception of self in the artificial reality")*.

This new discipline will also have to explore the possibilities and problems raised by the *multisensory interfaces* which can be implemented with today's (and especially tomorrow's) **multimedia systems** ([2], [21]).

Understanding the problems of kinestics and orientation in "strange" environments will be more and more important, also because the worlds represented by future workstations will not necessarily mimic the natural world according to the principles of **realism** and to the **laws of perspective**.

Nothing, in fact, prevents us from thinking of implementing "impossible" worlds, like, for example, those visualized in Escher's fantasies (Fig.15). Quoting again I.Sutherland (who built the first *head mounted display*, more than twenty years ago):

"A display connected to a digital computer gives us a chance to gain familiarity with concepts not realized in the physical world. It is a looking glass into a mathematical wonder land".

The possibility of creating, representing, exploring and changing, interactively, *"worlds that are not and never can be"* [10] suggests to our fantasy new, stimulating scenarios, from *interactive cinema* to *computer games*, to new forms of *computer-based-training*, to *computer art*, (see, for example, [49]) to the use of computer for interactive analysis of scientific phenomena.

The emerging discipline of *artificial life* ([26], [27]) will, in the future, help populating these worlds with simulated living creatures.

All this opens new and far-reaching possibilities to computer applications, not only for the development of new and more interactive apparatus for the *control of systems* (process-control, industrial automation, buil-

ding automation, ...), but also for the design of *information systems* of new conception:

"If artificial reality can be observed, then it is possible to enter in the artificial reality. If we can enter in the artificial reality, then we can have at our disposal all the real, made real/artificial close to us, wherever we are." [11]

11. ACKNOWLEDGMENTS

The Author is deeply indebted to Gianni Degli Antoni, to whom many ideas reviewed in this paper are due, who, in the last few years, has promoted the concepts of artificial reality by means of a large number of conferences, workshops, seminars, and research and prototyping activities.

Many thanks are also due to Roberta Macchi, for her invaluable activity within the HUG (Hypertext User Group) of the Dipartimento di Scienze dell'Informazione, University of Milano.

12. REFERENCES

[1] Abbott, E.A., FLATLAND - A ROMANCE OF MANY DIMENSIONS, First Edition: 1882

[2] Ambron, S., Hooper, K. (editors), INTERACTIVE MULTIMEDIA, Microsoft Press, pp.xi339, 1988

[3] Activision, Inc., THE MANHOLE, software for Macintosh, by Robin and Rand Miller, 1988

[4] Activision, Inc., COSMIC OSMO, software for Macintosh Robin and Rand Miller, 1989

[5] Apple Computer Co., HUMAN INTERFACE GUIDELINES: THE APPLE DESKTOP INTERFACE, Addison-Wesley Publishing Co., pp.xii 144, 1987

[6] Apple Computer, HYPERCARD, software for Macintosh, by B.Atkinson, 1987

[7] Bolt., R.A., THE HUMAN INTERFACE - WHERE PEOPLE AND COMPUTERS MEET, Lifetime Learning Publications, Belmont, CA, pp.xvii113, 1984

[8] Broderbund Software, WHERE IN THE WORLD IS CARMEN SANDIEGO, software for Macintosh, Apple IIGS, Amiga, by G.Portwood and L.Elliott, 1985-1989

[9] Broderbund Software, SIMCITY, software for Macintosh by Maxis Software, 1989

[10] Brooks, F.P.jr., GRASPING REALITY THROUGH ILLUSION - INTERACTIVE GRAPHICS SERVING SCIENCE, CHI '88 Conference Proceedings, Special Issue of the SIGCHI Bulletin, ACM Press, pp. 1-10

[11] Degli Antoni, G., IL COMPUTER, IL REALE, L'ARTIFICIALE, in Note di Software n.41, October 1988

[12] Degli Antoni, G., FROM REALITY TO HYPERMEDIA THROUGH ARTIFICIAL REALITY, in HUG89, pp.10-15

[13] Degli Antoni, G., ARTIFICIAL WORLDS, in HUG89, pp.8-9

[14] English, W.K., Engelbart, D.C., Berman, M.A., DISPLAY-SELECTION TECHNIQUES FOR TEXT MANIPULATION, IEEE Transactions on Human Factors in Electronics, HFE-8, 1967, pp.5-15

[15] Fairchild, K., Meredith, G., Wexelblet, A., THE TOURIST ARTIFICIAL REALITY, in CHI '89 Conference Proceedings, Special Issue of the SIGCHI Bulletin, ACM Press, pp.299-304

[16] Fisher, S.S., TELEPRESENCE MASTER GLOVE CONTROLLER FOR DEXTEROUS ROBOTIC END-EFFECTORS, in Proceedings SPIE - vol. 726, "Intelligent Robots and Computer Vision", 1986, pp.396-399

[17] Foley, J.D., INTERFACES FOR ADVANCED COMPUTING, in Scientific American, October 1987, pp. 127-135

[18] Gardin, F., LENTI, LEVE E SPECCHI SUL REALE E L'ARTIFICIALE, in Proceedings of Confe-

rence "Verso la comunicazione elettronica", Grande Fiera d'Aprile, Milano, April 20-21, 1989, pp.21-24

[19] Hammond, N., Allinson, L., TRAVELS AROUND A LEARNING SUPPORT ENVIRONMENT: RAMBLING, ORIENTEERING OR TOURING?, in CHI '88 Conference Proceedings, Special Issue of the SIGCHI Bulletin ACM Press, pp.269-273

[20] Hinton, C.H., A PLANE WORLD, in *"Scientific Romances"*, First edition: 1888

[21] SISTEMI INFORMATIVI E MULTIMEDIALITA', Universita' di Milano, Hypertext User Group, Proceedings of a Seminar organized by SOI Informatica, Milano, 9 November 1989

[22] IBM, SYSTEM APPLICATION ARCHITECTURE, COMMON USER ACCESS: BASIC INTERFACE DESIGN GUIDE, doc.no.SC26-4583-0, pp.xx 268, First Edition, December 1989

[23] IBM, SYSTEM APPLICATION ARCHITECTURE, COMMON USER ACCESS: ADVANCED INTERFACE DESIGN GUIDE, doc.no.SC26-4582-0, pp.xvi 195, First Edition, June 1989

[24] Infocom, Inc., THE HITCHHIKER'S GUIDE TO THE GALAXY, software for Apple II, PC, Macintosh and Commodore 64/128, by D.Adams and S.Meretzky, 1988

[25] Infocom, Inc., QUARTERSTAFF - THE TOMB OF SETMOT, software for Macintosh by S.Schmitz and K.Updike, 1988

[26] Langton, C.G. (ed.), ARTIFICIAL LIFE, Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems - Los Alamos, New Mexico, September 1987, Addison-Wesley Publishing Co., pp.xxix 655, 1989

[27] Langton, C.G., ARTIFICIAL LIFE, in [26], pp.1-47

[28] Lippman, A., MOVIE-MAPS: AN APPLICATION OF THE OPTICAL VIDEODISC TO COMPUTER GRAPHICS, Computer Graphics 14 (3), pp.32-42, 1980

[29] McGath, G., COMPUTE! GUIDE TO ADVENTURE GAMES, Compute! Publications Inc., Greensboro, North Carolina, pp.viii 203, 1984

[30] Microsoft Corp., FLIGHT SIMULATOR, software for PC, Macintosh and other computers by B.A.Artwick, 1986

[31] Mindscape, Inc., THE COLONY, software for Macintosh by A.D.Smith, 1988

[32] Mindscape, Inc., DEJA VU, A NIGHTMARE COMES TRUE, software for Apple II, Amiga, Atari ST, Commodere 64/128, Macintosh and PC by ICOM Simulations, 1985

[33] Mindscape, Inc., UNINVITED, software for Apple II, Amiga, Atari ST, Commodere 64/128, Macintosh and PC by ICOM Simulations, 1986

[34] Mindscape, Inc., SHADOWGATE, software for Apple II, Amiga, Atari ST, Commodere 64/128, Macintosh and PC by ICOM Simulations, 1987

[35] Minsky, M. R., MANIPULATING SIMULATED OBJECTS WITH REAL-WORLD GESTURES USING A FORCE AND POSITION SENSITIVE SCREEN, ACM Computer Graphics, vol.18, n.3, July 1984, pp.195-203

[36] NeXT Inc., THE NEXT USER'S REFERENCE MANUAL, pp.450, 1989

[37] Open Software Foundation, OSF/MOTIF STYLE GUIDE, 1989

[38] Pizzi, R., THROUGH THE LOOKING GLASS: UNA METAFORA DELLA REALTA' ARTIFICIALE, Proceedings of Conference "Verso la comunicazione elettronica", Grande Fiera d'Aprile, Milano, April 20-21, 1989, pp.7-16

[39] Polillo, R., PARADIGMI DI INTERAZIONE CON REALTA' ARTIFICIALI, Proceedings of the CNR Workshop "Conoscenza per Immagini" held in Rome, 27-28 November 1989

[40] Shneiderman, B., DIRECT MANIPULATION: A STEP BEYOND PROGRAMMING LANGUAGES, in IEEE Computer, August 1983, pp.57-69

[41] Smith,C., Irby, C., Kimball, R., Verplank, B., Harslem, E., DESIGNING THE STAR USER INTERFACE, in BYTE, Vol.7, n.4, April 1982, pp.242-282

[42] Spectrum HoloByte, FALCON, software for Macintosh and other computers by G.Louie and M.Johnson, 1987

[43] Sun Microsystems, OPEN LOOK GRAPHICAL USER INTERFACE SPECIFICATION, 1989

[44] Sutherland, I., THE ULTIMATE DISPLAY, Proceedings of the IFIP Congress, vol.2, 1965, pp.506-509

[45] Tello, E.R., BETWEEN MAN AND MACHINE, BYTE, September 1988, pp.288-293

[46] Weimer, D., Ganapathy, S.K., A SYNTHETIC VISUAL ENVIRONMENT WITH HAND GESTURING AND VOICE INPUT, CHI '89 Conference Proceedings, Special Issue of the SIGCHI Bulletin, ACM Press, pp.235-240

[47] Weizenbaum, J., ELIZA, Communications of the ACM, 9:1, January 1966, pp.36-45

[48] Winograd, T., Flores, F., UNDERSTANDING COMPUTERS AND COGNITION, Ablex Publishing Co., pp. xiv 207, 1986

[49] Kruegerg, M.W., ARTIFICIAL REALITY, Addison-Wesley Publishing Co., 1983

A MODEL FOR ARTIFICIAL REALITY

Francesco Gardin
Dipartimento di Scienze dell'Informazione
Università degli Studi
Milano

RIASSUNTO

In questo articolo viene proposto un modello generale di realtà artificiale basata sul calcolatore.
E' operata una distinzione tra realtà artificiali basate sul calcolatore e simulatzioni della realtà basate su altre tecnologie.
Viene inoltre puntualizzata la distinzione tra realtà naturale, estensione/simulazione fisica della realtà naturale (Realtà Reale), simulazione software della realtà, Realtà Artificiale e sua integrazione con la Realtà Reale.
Viene anche discusso il contributo che la realtà artificiale può portare alle scienze cognitive.
Il modello proposto è incentrato sul concetto di agente, questi si suddividono a loro volta in agenti 'hard' ed in agenti 'soft', dove i primi appartengono al mondo reale e i secondi sono invece modelli software percepibili. Combinazioni tra due sono inoltre possibili. Gli agenti sono essi stessi suddivisi in due componenti: la parte percettibile e la parte comportamentale. I comportamenti sono a loro volta componibili.
La descrizione del modello è corredata da un numero di esempi ed un elenco di principi di progettazione di realtà artificiali, quali il principio di simmetria, di auto-percezzione, etc., che sono espressi mediante le primitive concettuali del modello proposto.

ABSTRACT

In this paper we propose a general model for computer-base artificial reality.
A distinction is made between computer based artificial reality and non-computer based technology simulations of reality.
A distinction is also drawn between natural reality, physical extension/simulations of natural reality (Real Reality), software-based simulation of reality, Artificial Reality and its integration with Real Reality.
The potential contribution of artificial reality to the study of cognitive science is briefly addressed.
The proposed model is based on the notion of agents: they can divided in hard and soft agents, where the former belong to the real world, and the latter are perceivable software models. Combinations of the two are also possible. Agents are themeselves divided into two components: the perceivable aspects and the behavioural aspects, Behaviour can be compositional.

The description of the general model is then followed by a number of examples and design principles for artificial realities, such as the symmetry and the self-perception principles, etc., which are expressed in terms of the conceptual primitives of the proposed model.

1. INTRODUCTION

During recent years, awareness about the importance of man-machine interfaces and the availability of multimedia technologies [3] has turned this topic into a central research issue, and the need for a unifying, basic paradigm has emerged.
We believe that the answer is represented by the notion of Computer-based Artificial Reality (referred to he-

reafter as Artificial Reality [5], unless otherwise specified), i.e. the ability to develop perceivable software models, as surrogates/extensions of reality, where users perform more and more tasks previously done in the real world [6], [8], [7].
Incidentally this has raised the issue of the distinction between artificial and real reality, or, what is the real world.

Although all sort of philosophcal, psychological and sociological implications follow, we didn't have to wait for Artificial Reality, made possible by computers, in order to witness instances of natural reality modified by technologies: cars, radios, microscopes etc. are good examples of surrogate/extensions of natural reality, where people can move around quicker, sound waves travel faster and things look bigger.
A theater stage is again a simulated reality (natural, or extended), which makes use of physical objects.

Computers make possible software simulation of real objects (or objects with no correspondence in the real world). Adding to simulation the possibility of being perceived not only numerically, but also graphically (e.g. scientific visualization), has given a different identity to specific simulations. These perceivable models can be regarded as objects. It is a new reality, artificial [5], in the sense that its elements are perceivable computer simulations.

The key element in Artificial Reality is one of a quantitative nature: our increased ability to model more and more complex objects/environments (both the behaviour and their perception) and the availability of computers powerful enough to run and display such models.

Software-based modelling, compared to all previous physical technologies, has put into the right perspective the concept of physical-based extensions/simulation of the natural reality performed so far.

More generally, the whole human perception of reality is under discussion as a result of this. As for other revolutions, by shifting the center of reference from reality, (before the advent of computers), to the whole universe of possible realities as they can exist today, we might appear to have completely lost the reference.

On the contrary, our ability to perceive has become the central factor, since human perception has lost its unique relationship with the usual reality, where perception is intimately related with what is perceived, due to the uniqueness of reality. (Despite the fact that special physical devices have been implemented to deceive human perception, e.g. stroboscopic lights, the order of magnitude makes the difference.)
We now have maximum freedom to design and implement all sorts of worlds. The cognitive factor is therefore the fundamental issue.

Therefore, by trying to propose a model of Artificial Reality, where existing man-machine interface paradigms [2] and applications are its corollaries, we think we are indirectly giving a contribution to the understanding of human cognition [6]. The principle of self-perception described later in the paper is an example of this aspect.

As for Artificial Intelligence, both the shifting from human intelligence to intelligence *per se*, and the generalization of human intelligence, has been made possible by the existence of the computer as the most powerfull modelling tool mankind has ever had, and has helped to clarify the distinction between intelligence and human intelligence as a specific instance of the former.
All this raises the question of what the relationship is between Artificial and Real reality.

The generalization of perception (the cognitive factor), makes a unified view of artificial and real reality possible in terms of the universe of possible perceivable realities.

In order to model the above unifying view, we introduce the concept of the agent and its breakdown into perceivable and behavioural components. Agents can be hard agents (existing physical objects) and soft agents (computer-based models).By mixing the two aspects of hard and soft agents the two become part of the same extended reality. By doing so we can create a bridge between the two, so that we can naturally move in and out from Artificial and Real reality.

Essential to this bridging is an intermediate model

whose purpose it is to make the hard agent perceivable by the soft agent and vice versa.
So, for example, in the case of a keyboard (the model of the user from a soft agent stand point), the intermediate model is represented by a sequence of ASCII characters, while in the case of the printer (the model of the soft agent from the hard agent stand point), the intermediate model is represented by the sequence of printable characters and the 'x' and 'y' printer's head movement directions.

Although the role of the intermediate model could be hidden as part of the hard or soft agent model, it can be considered either as the key perception element of the target agent, or a property of the origin agent. The difference will be explained in details in the paper.

It is also worthwhile to notice the increasing important process of 'being absorbed' into the artificial world generated by the computer. This one of the most dramatic changes taking place at this very moment. More and more complex computer based realities, and merges between real and artificial realities, are being developed with the consequence that more and more tasks are performed within these realities as opposed to the real world. We need to understand in depth how these realities should be developed, related to real reality and used to amplify our capabilities.

In a sense this is the basis of a major biological evolution, because of the radical change in the environment in which we will live. The rate of this change and its order of magnitude is growing very rapidly: proportionally to the MIPS of our computers and our modelling ability.

The effects of the corresponding evolutionary process are not known, since we don't know which artificial realities will occur in the future, but new opportunities will be created as for any environmental change.

Living in a world where the laws of ordinary physics are no longer true, will cause a number of fixed points in the current reality to disappear: forms will be no longer as static as they are today, for instance. Just the consequences of this change would be enough to justify a major change in human communication.

These points will not be developed in this paper, but are issues which deserve our attention if we want to understand their effects.

2. NEED FOR A MODEL

Designing artificial realities can only be turned into a professional activity if we can develop a model of such a modified/extended version of reality. Such a model represents the reference point for a coherent approach to apparently different situations. This is only possible if we can capture the essential features of an artificial reality.
As for any model, we expect that it represents not only a guideline for the designer of man-machine interfaces, but also a predictive tool for the analysis of cognitive processes related to the use of an interface for given tasks, or the consequences of a set of initial design assumptions of the interface.

The starting point in our definition of a model,is the notion of agent, broken down into its perceivable and behavioral aspect (fig. 1a).
An agent can be either real: hard agent (fig. 1b), or artificial: soft agent (fig. 1c), so for example, a user, a printer, a chair and the computer itself are real.
An artificial agent is any perceivable software-generated behavior which exists only as a result of an active calculation by a computer, with its own identity over time. Examples of artifical agents are a word processor, an icon, a simulated bouncing ball, a synthesized sound, etc.

Even though the above classification can lead to a number of ambiguities, in our view it is a convenient zero-order approximation. Some of the potential problems of our model will be discussed elsewhere in the paper (e.g. the border-line between perception and behavior is very subtle), for the time being the notion of hard and soft agents is accepted as correct.

3. THE MODEL

The basic components of an Artificial Reality are agents. Agents are the combination of two components: the generation function (in charge of behaviour) and the perceivable parts. Although this is a very approximate

distinction and a gray area divides the two aspects, we assume it is acceptable for the degree of approximation of the proposed model.
Moreover agents can be divided into real agents, soft agents, or some combination of the two.
A few examples should help to clarify our definitions. A real car, according to our distinction, has a generation function represented by its dynamic and kinematic properties (max speed, acceleration, max steering angle, etc.) and the perceivables, such as its shape, color/s, etc.. The car described so far is a real agent (fig. 1b). A soft car (soft agent, fig. 1c), could be a car whose dynamic and kinematic properties are defined according to some non-real physics: acceleration as a step-like function (e.g. from 0 to 100 km/hour instantenously), penetrability with other bodies, etc., whereas its perceivable aspects are non-existent (i.e. the car behaves according to the dynamic and kinematic defined model but cannot be perceived by human beings), or can be perceived only every ten seconds, or each time the speed is above 130 km/hour and its colour changes, or its shape turns from a turtle into a rabbit. Cartoons cars are a good example.

A hard/soft car is a mixture of the two, where the perceivable aspects are generated by computer, and the generation function is performed by the real car. This is a typical example where a real car, whose coordinates are detected through radio signals, is then represented on a computerzied city map showing its position on the map (fig. 1d). Cars exceeding the speed of 130 km/hour will start to flash, in order to attract viewer's attention. Clearly the mixing could be exchanged, namely the generation function could belong to the soft agent and the main feature of the real car could be only its perceivable aspect. Although it is less intuitive to think of a sensible example of such combination, we expect that in some suitable universe such a car exists. Indeed one particular combination of such a mixture belongs to a very well known universe, namely the world of car simulators (similar to flight simulators, fig.1e), where a driver sits in the car and the external environment is projected outside, so that the effects of the soft generation function properties are responsable for the changes in the external views (e.g. the inertial properties of a real car neglegible, so that 90 degrees changes in trajectory are possible, even at very high speed, without affecting the stability of the car), or are responsable for some specific tilting or shifting of the body of the car. The two effects are complementary, as for relative motion in cartesian geometry: either one moves his/her own point of reference, or somebody else's; the final effect is the same: motion is perceived.
Different hard/soft combinations, requiring the use of effectors, could be more difficult to think of and implement.
In general, when hard and soft components are mixed, sensors and effectors are required to relate the two.This brings into the initial model, which represents the first order of approximation, the first order of approximation, namely, the intermediate model (filter).
A further corollary of the proposed model is the fact that agents, either software or hardware, can have more than one perceivable aspect at the same time, but only one in the same time-space unit (fig. 1f).
So, for example, a real car has its own perceivable aspect in the real world (specific space-time unit), but the very same car can have a different aspect in the space-time unit of the computerized city map monitor. The same principle holds also for the behavior aspect, in the sense that two different behaviors, can be combined into a single behavior, or a single perceivable (fig. 1g). So, for example we can couple the behavior of two users to produce their combined behavior, or just the same perceivable. This example also shows how ambiguous the distinction between behavior and its perceivables is.

4. MORE ABOUT THE MODEL

The behaviour/perceivables model is infact rather simple, and couldn't explain for example the 'straight line drawing case'. The following example is used to introduce the notion of intermediate model (filter), fig. 2, and therefore to prepare the ground for introducing the subtle cognitive implications of Artificial Reality, which can be partially extracted by the principles described later in the paper.
The example itself is about the 'perfector' property of an Artificial Reality. When the mouse is used to draw a line within a line editing program, a straight line option can be selected which allows the user to draw a straight line, despite the 'tremulous' of his hand dragging the mouse. How do we explain this process within the

proposed model of artificial reality ?

There is a hard agent, the hand, together with the associated user, and there is a straight line being displayed. The line is not just the projection of the hard agent into the artificial reality, since the hard agent cannot generate the behaviour which would result in the drawing of a straight line: something must be missing. What is missing is the filter (the intermediate model), which turns an irregular line into a straight line (the process is actually much more complex since it is generally incremental and the movement of the hand is governed by the feedback from the artificial reality, rather than the usual feed back from the same reality where the action is taking place: the Real Reality).

This filter is the actual behaviour for the perceivable component of the straight line in the artificial world. In general each 'in' and 'out' passage from real to artificial and vice-versa must involve some model. The difference between the user with and without the linearizing filter, is the difference in the model that the artificial world perceives of the user: a sequence of couples of digits in both cases (x and y coordinates), but with the linearizing filter the sequence must obey very specific rules (each couples of digits follows, or is followed by, a couple so that both are on the same straight line).

Does the filter belong to the hard agent or somebody in the artificial world ?

In other words is it a way to design super users (who can draw perfect straight lines), or super worlds where, under certain conditions, only straight lines can be drawn ?

We could use both views, but for consistency reasons let's assume that this capability belongs to the hard agent, since every time we admit input from the real world we must allow for some transformation, both physical and behaviouristic). The straight line is therefore the result of the combination of two behaviours: the hand and the filter. In such a case we are referring to it as the behavior functional application, that is the final behavior can be viewed as the application of a function (the filter) to the application of another function the behaviour of the user hand (the domain is represented by the hand commands and the co-domain is the result of the mouse movements).

We have decided to attach the filter to the user, although the filter is actually computed by the machine, on the other hand the filtering function could also be computed by an on board computer in the mouse.

Symmetry holds: for actions performed on real reality (by means of actuators), where the filtering problem applies, in the reversed way (from Artificial Reality to real reality), the filtering model has decided it will belong to the soft agent.

If we thought that the above would solve some of the problems we might encounter in the use of the model, let's consider the case where the straight line is not drawn in the artificial reality, but directly on real reality, e.g. when we use the mouse to draw a straight line on a piece of paper, by guiding a robot holding a pen. In such a case, in order to be consistent with our definitions, we admit the presence of two agents: a hard agent who draws straight lines in an artificial world, and a soft agent which draws straight lines in the real world. Again, for consistency reasons the super agent is the user, who can draw straight lines not only in the artificial world, but can also give instructions to a software agent to draw straight lines in the real world.

Controlling a robot through an Artificial Reality, or in an Artificial Reality, is a very good example of cognitive body dissociation. Again driving a car is essentially the same thing, what makes the difference is the order of magnitude which can be achieved in the dissociation, which can give rise to phenomena that otherwise would probably be undetected.

5. DESIGN PRICIPLES FOR ARTIFICIAL REALITIES

The model described so far, has a number of advantages. Two of these are: a notation for describing Artificial Realities, a useful framework to explore and state design principles for such realities.

Similar to architecture and any other design activity which has been around long enough to form a consensus about how things should be done (principles) [1], Artificial Realities, in the form of computer interfaces, are already in a position where a number of elementary design principles have already emerged.

A classic text about design principles for man-machine interfaces in engineering is [9], a good example of how text books for Artificial Realities engineers and architects, will be.

The design principles briefly described below, are the

result of an analysis carried out on existing successful computer interfaces in terms of the language of the proposed model.
The reason why we used successful computer interfaces, is based on the assumption that if something is used successfully (i.e. freely choosed by users) by hundreds of thousands of people it captures some basic properties of the users (indirect/reverse engineering). For example we could study some features of human body through the carefully analysis of a driver's operating environment in a car interior.
Similarly, if pull-down menu's appear to be more successful than line-oriented text commands, this gives an indication of how limited our capability is to remember text commands (words in general), if they are not used very often.

It is not our intention here to explore this aspect of computer interfaces, i.e. relate their success to users' features; our objective here is instead to show how the simple proposed model can be used to describe design principles used in computer interfaces. It should be interesting to notice that some of the design principles listed below have been implemented in existing systems but perhaps they are not the result of a deliberate analysis, but more something which turned out to be necessary at that point in the implementation of the system.

P1. Principle of Common Space for Interaction

This principle is about the common time-space unit. It states that any two agents, in order to interact, must be able to share a common space, either hard or soft. According to this definition, a user can write a letter on a piece of paper, or if the system used is a word processor, he will have to trasform himself into the soft agent who moves around writing sequences of characters in the Artificial Reality of the soft letter. In both cases the interaction requires the two agents to come into contact in the same hard or soft space.

P2. Priciple of Symmetry

The principle of symmetry states that in the design of an interface the relationship between any two agents is symmetrical. By relationship, we mean the capability of one agent to perceive and be perceived, or act and be reacted to. Whenever the symmetry doesn't hold the relationship is unbalanced and must be looked after carefully.
As an example we can use the relationship between the user and a word processor. The user *per se* is a hard agent, the word processor, on the other hand is a soft agent. In order for the user to access the word processor he/she must be able to see it, namely it must have a perceivable aspect. On the other hand, in order for the word processor to be used by the user, he/she must be perceivable by the word processor. In general there is little difference to the word processor whether the agent on the 'other side' is a human being or a cat walking on the computer keyboard.
What matters in the end is the fact that the behavior generator (resulting in a sequence of text/commands), will be perceived as a correct string of ASCII characters. (It is interesting to note that a pre-coded sequence of text/commands could have been stored on a file and used for the word processor, it is arguable to say whether the user behavior aspect is soft, or hard; the time-scale is crucial for the answer).

P3. The principle of self perception X.3

Any agent must be able to perceive (him/her)self in the same form as he/she perceives the agents he/she is interacting with. This is a typical example of the case where the interaction with an artificial reality has clarified the role of self perception, that is the continuous feed-back between the *se* and the perceivable aspects of the *se* (the body). Acting in real reality is quite natural, therefore we take for granted the fact that we perceive ourselves. On the other hand, when we deal with an artifical reality there is a dichotomy between us and our body, since our body becomes whatever we use to act on reality or Artificial Reality; and we need to perceive our body to close the continuous feed-back, in order to control our acts. The perceivable aspect of the mouse (e.g. the arrow is displayed) is the condition for us to be able to perceive ourselves and therefore control our actions.
The alternative to self perception, is memory: if one can remember every action exactly one doesn't need to see where one is. Apparently nature has adopted a solution where the world acts as a memory, and all we have to do is take a look at it, rather than remembering it.

P4. Principle of minimum effort, or maximum performance

According to the principle of minimum effort, or maximum performance, each time an Artificial reality is designed, the filter (in both directions), should be conceived so that the effort to perform an action is minimized, or, conversely, the maximum performance is achieved (e.g. the straight line case described earlier).
Whether the filter belongs to the hard agent, or the soft agent is an open issue which can be decided only as a matter of convention.

P5. Priciple of continuity, or spacial positioning

This principle again concerns perception. The underlying assumption of our perceptual system is such that we expect physical objects to maintain their position in space, unless some specific action is performed on them, as well as to react to actions in such a way that continuity in movements is fulfilled.
The result in an Artificial reality which doesn't take into account this principle is cognitive disorientation.
This principle can be expressed in terms of the primitives of the proposed model as constraints that we add by default to the behavior of soft agents and corresponding filters.
Sometimes it could be convenient to design Artificial Realities which are not based on this principle (Artificial Reality doesn't need to be a replica of Real Reality), but this should be a deliberate choice.
Icons which disappear when they are dragged around with the mouse, and re-appear in the final position when the dragging is completed, go against the principle of continuity.

P6. Principle of causality in Artificial Reality

The principle of causality is about the physics of Artificial Reality, and the effects that it causes on perception. The design of soft agents should be aware of the physics they obey.
A word processor is a good example of an Artificial Reality where ordinary laws of physics no longe hold. The amount of energy required by a user to erase a line of text is the same as for a page of text. Similarly, moving a block of text upward or downwards takes the same amount of effort: i.e. a word processor is not subject to gravity.
The absolute freedom in designing a new world which doesn't necessarily obey the laws of physics, is on one hand a great opportunity, and on the other hand a major cognitive challange, since all sorts of nasty effects could follow. Acting in a world where objects don't penetrate each other represents a source of greater predictability than a world where everything can be penetrate everything else.
A set of rules to check the consistency of the physics used, with respect to the cognitive effects on causality it can produce, could be added by default both to the soft agents and the filters.

The above list of principles is by no means exhaustive, but it is an indication of notions which should be taken into account when designing an Artificial Reality.
They can be expressed in terms of the primitives of the proposed model: soft and hard agents, behavior and perception parts and filters (intermediate models).
We leave the reader, as an exercise, to revisit familiar Artificial Realities (e.g. word processors, iconic interfaces, CAD systems, Flight Simulators, Computer Games, etc.) in terms of the above primitives and principles.

7. FINAL REMARKS

As mentioned earlier, more and more objects and related functions which used to exist only in the real world are now being implemented as computer models. Word processors, CAD systems, electronic mail, icon-based calculators and watches, video games, etc. are just a few examples of objects which we use increasingly frequently in our lifes. Our capability to design and implement models of more and more real objects (or their extensions, or just completely new ones) will increase such a trend even more.
We are already in a situation where most ordinary office operations could be carried out in an artificial reality running on a workstation.
On one hand the opportunities for the creation of entirely new worlds is fascinating; on the other hand, the design of an artificial environment, as for any design activity, needs a deep understanding of the properties of the environment itself, the user and how operations are

performed either by the user in the environment or among several users.

Existing man-machine interface paradigms (button-based, pull-down menus, etc.) cannot be only used for rich and complex environments, where a number of criteria taken into account in the design of the real environment where we live and work are completely left out.

We are aware of the fact that perhaps its is possible to produce counter-examples in the proposed model, but the need of a model, even if approximate (a higher order of approximation could be added too), represents a priority at this stage of the development of Artifical Reality, as a discipline.

7. AKNOWLEDGEMENTS

The author wishes to thank Gianni Degli Antoni, Jaron Lanier, Flavio Argentesi for the discussions and the support received so far.

9. REFERENCES

[1] C. Alexander, S. Ishikawa, M. Silverstein, A PATTERN LANGUAGE, Oxford University Press, New York, 1977

[2] R. Bolt, THE HUMAN INTERFACE: WHERE PEOPLE AND COMPUTERS MEET, Wadsworth, Inc., Belmont, CA 94002, USA, 1984

[3] S. Brand, THE MEDIA LAB:INVENTING THE FUTURE AT MIT,the Penguin Group, Viking Penguin, USA, 1987

[4] S. Card, P. Moran,A. Newell, THE PSYCHOLOGY OF HUMAN-COMPUTER INTERACTION, Lawarance Erlbaum Associates, Inc., Hillsdale, New Jersey, USA, 1983

[5] G. Degli Antoni , ARTIFICIAL WORLDS, in UHG89, pp.8-9, 1989

[6] F. Gardin, LENTI, LEVE E SPECCHI SUL REALE E L'ARTIFICIALE, in Proceedings of Conference "Verso la comunicazione elettronica", Grande Fiera d'Aprile, Milano, April 20-21, 1989, pp.21-24

[7] F. Gardin, G. Haus, I. Macchia, G. Molendi, A PROJECT ON SOUNDS IN ARTIFICIAL REALITY, in Note di Software 48/49, 1990

[8] Krueger, M.:"Artificial Reality", Addison-Wesley Publishing Company, USA, 1983

[9] M.S. Sanders, E. McCormick, HUMAN FACTORS IN ENGINEERING AND DESIGN, McGraw-Hill Book Company, New York, 1987

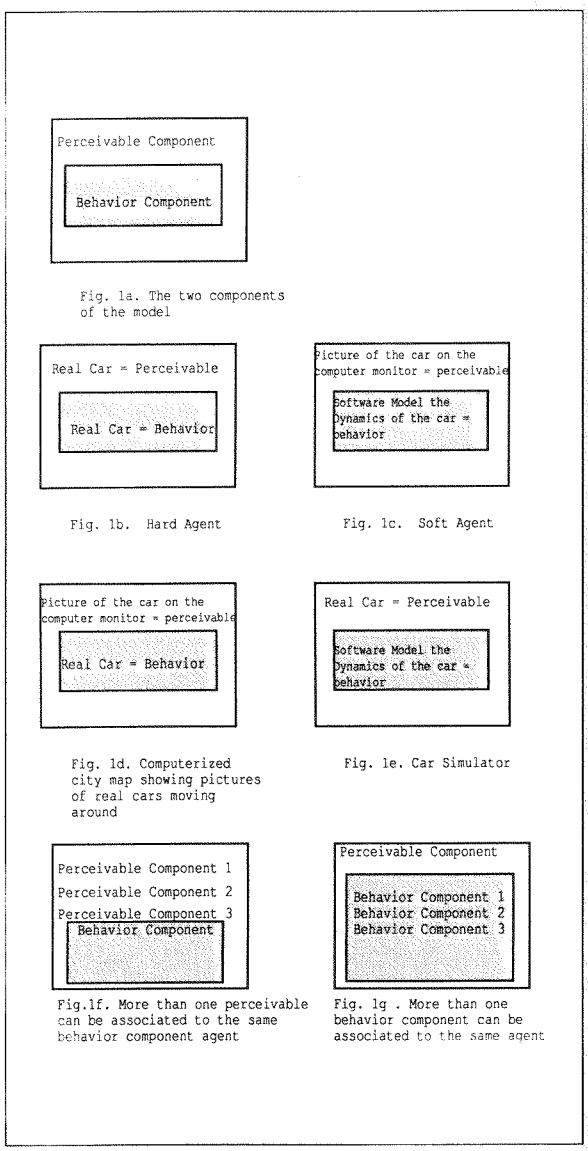


Figure 1:

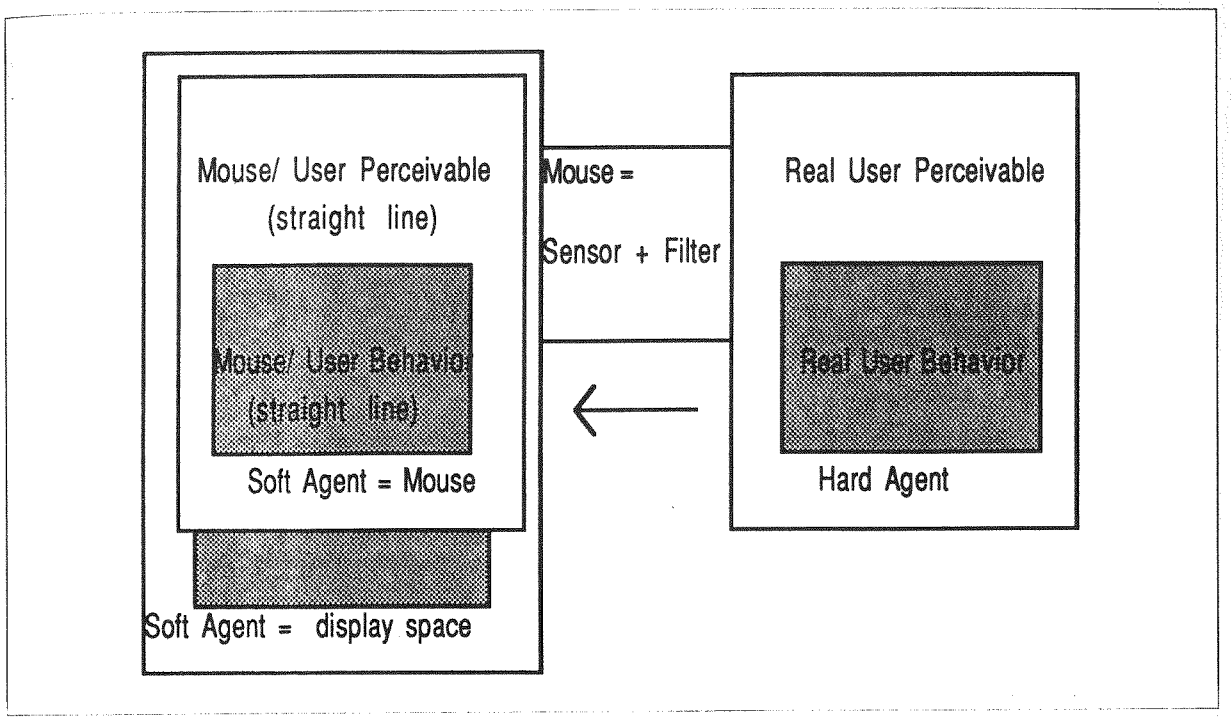


Figure 2: Need for a filter to allow hard and soft agent to interact. By replacing the sensor with an actuator the direction of the arrow can be reversed, e.g. printing a line

SYMMETRY IN MAN-MACHINE COMMUNICATION

Roberto Grande
Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano

RIASSUNTO

In questo articolo viene riconsiderata la comunicazione tra uomo e computer alla luce di un promettente paradigma: la simmetria. Prendendo spunto dalle conversazioni tra persone sono confrontati i ruoli che assumono gli esseri umani nelle loro interazioni e nelle interazioni coi computer. Da qui si suggerisce che l'analogia conversazionale vada estesa per includere la simmetria nella comunicazione. Si propongono quindi tre proprietà che devono essere soddisfatte da una comunicazione simmetrica e che riguardano i suoi aspetti sintattici, semantici e pragmatici. Come esempio di un primo passo verso l'applicazione di questo paradigma viene presentato ProTalk: un ambiente di programmazione basato su Prolog in cui interprete ed utente adottano un linguaggio di comunicazione comune ed un modello uniforme di pragmatica della comunicazione. Entrambi possono inviare e ricevere conoscenza su un determinato problema allo scopo di sviluppare, in modo incrementale, un programma Prolog che lo risolva. Infine viene suggerito come il paradigma della simmetria permetta di ridurre la tradizionale separazione tra utente finale e programmatore.

ABSTRACT

In this paper, man-machine communication is revisited from the point of view of a promising paradigm: symmetry. Starting from the analogy between person-to-person conversations and computer-to-person interactions the roles played by person and computer are analyzed. It is then suggested to adopt the paradigm of symmetry in order to achieve a more conversational attitude in computer interactions. Three properties are proposed as guide-lines for such a paradigm, concerning communication syntax, semantic and pragmatic. As a first step toward symmetry ProTalk; a Prolog based programming environment, is presented in which the interpreter and the user communicate by means of a common language and in accordance with a common pragmatical model. Each part can send and receive knowledge concerning a given problem, thus incrementally developing a Prolog program that will solve it. Finally, it is suggested that the paradigm of symmetry could provide a way to blur the separation between end user and programmer.

1. CONVERSATIONS WITH COMPUTERS

Nowadays the role of computer as a communication medium has reached wide acceptance. People use computers to communicate ideas, data, experience, knowledge over time and distance, making them widely available to other people. They exploit the "reactive" nature of computer to encode and disseminate not only passive information but active working knowledge with which the recipient can interact.

We take the historical view of computers as mere "computational engines" being a lumpy case in the previous scenario: at the beginning, computer-mediated communication was mainly intended to share results (and operations by which these were attained)

within a restricted working group, regardless of the wealth of communication styles.
After the advent of the first interactive systems and the diffusion of personal computers, it has become common to speak of "conversational" interaction with a computer [1], [2]. This locution was intended to suggest the analogy between person-to-person conversation and computers-to-person interaction. The relevant point of the analogy is that communication with a computer program is somehow similar to communication with a person (for instance the person who wrote it) because it is two-way and interactive.
Computer scientists have not developed the conversational metaphor systematically; they started with the first attempts of emulating the human conversational capabilities, arising from the artificial intelligence arena [3], thus focusing on the technological and ergonomical aspects involved in computer-mediated communication. We can recognise the "direct manipulation" paradigm of current User Interface and their multimedia features among the most remarkable achievements of that effort.

2. THE GAMES PEOPLE (AND COMPUTERS) PLAY

An underlying assumption has characterized the most developments in this area: partners involved in the interaction play a game with assigned roles[1] . The user "uses" (thus his name) the active knowledge encoded in a program, the program delivers it, under the user's control, in order to accomplish some (often unrecognizable) of the user's intents.
Even when interplaying with those programs that exhibit a more conversational attitude, like Expert Systems (reminiscent of former Question Answering Systems), we can recognise the same communication pattern coming from an explicit roles separation. What characterizes these last programs is a capability of acquiring new knowledge during user interaction; but throughout the separated phases of knowledge acquisition and consultation, the roles of each partner (as knowledge receiver or knowledge provider) still remain fixed.
Moreover, during the acquisition phases, the user/expert conveys new knowledge in a purely syntactical form, as passive information, without any advice about the live nature it will assume when delivered by the program. In the previous schemata it might be convenient to force different communication styles that can best suit specific assigned roles, though this is far away from what we can observe in everyday conversations. Here partners' roles cannot be assigned in a predetermined way but they are frequently exchanged depending on the course of the interaction: each partner has the same chances to play as knowledge provider or knowledge receiver on the base of his own competence. This flexibility typically results in a fine granularity[2] of the interaction without hampering its overall effectiveness; indeed true conversations do not require a heavy context switching like that we find with a typical Expert Systems moving from knowledge acquisition to consultation and back again.

3. COMMUNICATION SYMMETRY

Because of the above attitude in assigning roles, we are not surprised if a leading aspect involved in person-to-person conversation is not adequately covered by the computer-to-person counterpart. We refer to the prominent symmetry exhibited in human conversations.
As a natural evolution of an idea first suggested by G. Degli Antoni in [4] we propose, in the following, a simple guide-line for introducing symmetry in man-machine communication.

Concerning the communication between two agents we will state its symmetric nature on the base of three outstanding properties:
a) (syntactical) the two agents share a *common language*
b) (semantical) they have equal capabilities to *attribute meaning* to each construct of this language[3]
c) (pragmatical) they have equal capabilities of *intervening properly* in the communication.

The first property requirement to both agents is a common syntactical skill. The last two need a shared background knowledge enabling them to supervise the interaction. In a more formal setting we should regard such background knowledge as "metaknowledge" because it deals with the specific-domain knowledge object of the communication[4]. This metaknowledge specifies how to relate the language's constructs to previous (encoded) knowledge, how and when to pro-

perly ask for missing knowledge, how and when to provide requested knowledge[5].

Specifically the word "properly" occurring in the third statement refers to the requirement that both agents conform to a common *pragmatic model* of the communication (i.e. a protocol) as it has been analysed by Levinson [5]. This model structures the interaction as an alternation of communication acts, each one belonging to a couple of complementary acts for instance of the kind *Question / Answer*. In each couple the former is referred to as an *opening* act and the latter as a *closure* act; each opening act should be followed, sooner or later, by its closure. We can have nested couples of communication acts to any depth level; these couples should be correctly managed by mean of a LIFO schemata. So attempting to answer a given query of one agent a further question might arise and so on , in turn, until all the open queries, starting from the innermost, are closed by an appropriate complementary answer.

4. WHY COMPUTER CONVERSATIONS LANGUISH

Symmetry in human conversations is so natural as to go unnoticed. It is demanded by the highly cooperative nature and the fine granularity of such interactions.
On the other hand, human-to-computer communication, still being two-way and interactive, could not be considered really conversational because of its evident asymmetry.

A simple means to judge the symmetry of such "artificial conversations" could be borrowed from the Artificial Intelligence side, adapting the well-known "Turing Test": a communication between two agents, say Human and Computer, might be considered symmetrical if we are not able to distinguish them just by looking at the form (not at the content) of their communication act.

More analytical (and perhaps more constructive) judgements could be derived by attempting to apply the three basic properties stated above.
A primary source of asymmetry can be recognized in the deficiency of the property a). In fact, during the same interaction, humans and computer programs communicate using languages that have different syntaxes and, often, different sets of symbols.
Moreover these languages cannot be translated one into the other on the basis of a common set of actions or a shared bunch of meanings to refer to. Depending on the roles played by human and computer in a specific interaction context, the sets of actions each one can undertake and the kind of knowledge they can exchange are significantly diverse from one side to the other. This also wipes out the second property.
It is difficult to state a general settlement for the last property. It could be roughly equated to the degree of interactivity enabled by the program, thus varying from great freedom (e.g. in an interactive computer game) to a strictly predetermined sequence of steps (e.g. in the *install* phase of most packages or interacting with a typical programming environment), to the coexistence of both tendencies in the same interaction (e.g. remind the occurrence of "modeless dialogues" and "modal dialogues" in a typical Macinthosh session).

At this point a very pragmatic question could arise: what kind of advantage should derive from imposing a conversational style, and specifically symmetry, on interaction with a computer program? Is it just an aesthetical concern? Or the is reason that we enjoy conversations more than working with unresponsive computers?[6]

5. THE EXAMPLE OF A PROLOG PROGRAMMING ENVIRONMENT

In the following we want point out the advantages that could be gained by adopting a more conversational attitude in the usual task of programming by dint of symmetry.
Let us consider programming as a special case of problem solving. The programmer is concerned with a task more comprehensive than just the implemention of an already fully specified algorithm; moreover he has to deal with the refinement of formal specification suitable for the *intended meaning* of the program: i.e. *what* the program is intended to serve for and *how* it is intended to accomplish that goal.
Deriving executable code from a complete and correct set of formal specifications could be regarded as a *transformation* task (and then it could be automated once and for all), whereas the specification phase requi-

res true problem solving skills [6].
It is mainly in this phase, we believe, that symmetrical interaction with the programming environment might enhance the programmer's problem solving capability by cooperating towards a satisfactory specification of the problem.
To demonstrate the previous idea a prototype of an interactive Prolog programming environment called ProTalk has been developed that realizes a first step toward symmetry.
The choice of Prolog as test-bed was determined by several reasons. Logic Programming enables a *declarative* technique of specification which is very suitable for the descriptive style of thinking typical of most problem solving tasks. As a practical expression of the Logic Programming paradigm, Prolog provides an effective way to completely execute such specifications [7]. Furthermore, from a pretty communicational point of view, logical languages come closer than other "artificial languages" to one of the most natural forms of human utterance: natural language. All these considerations suggest that Prolog could be regarded as good candidate for putting man and computer in touch as far as it is quite natural and still executable.
In spite of the above pleasant characteristics, actual Prolog programming environments do not exhibit a more evident inclination to conversations than conventional programming environments.
Before showing the peculiarities of ProTalk, it is worthwhile considering the typical scenario in which Prolog programs are developed, looking for symmetry.
The programmer and the programming environment share (to some extent) a common syntactical skill in Prolog.
The programmer might rely on the declarative semantics of the language (almost in the first phase, dealing with more operational concerns in a further optimization step). The programming environment, on the other hand, relies on the operational semantics provided by the Prolog abstract machine encoded into the interpreter.
These two ways of attributing meaning to the same language have been demonstrated to be equivalent as far as we can cut off the effect of some "extralogical" constructs [8]. On this ground (and assuming a sound implementation of Prolog) we are granted what the programmer recognizes as lo*gical consequence* of some previous statements, the programming environment is able to *deduce* from those same statements, and vice versa.
Till now the first two properties required by symmetry seem ready to be achieved.
What is unlikely to be achieved, without a radical reconsideration of present environments, is an effective interactivity. In this context this means enabling both the programmer and the programming environment to intervene at each step of the specification process contributing to the correct and complete development of the program.

The communication pattern we observe during interaction with a typical Prolog interpreter is as follows:
- from the *programmer* side:

* load (*consult*) a previously edited program into the knowledge base
* ask a query to the interpreter

- from the *interpreter* side:
* answer *yes* , reporting the successful variable substitution, or *no* , on failure.
* prompt the programmer for a new query

When the interpreter is set to the *debug* mode it essentially provide a more verbose account of the *yes /no* answer. In the best case the programmer can interactively ask the debugger to unfold or skip some parts of that account, but he cannot interactively modify the program and then resume its evaluation.
In such a scenario there is no means to verify "in itere" that the program under development behaves the way we want, neither to have some timely hint about what is missing or wrong. Indeed, whereas it is often preferable to specify a Prolog program in a *top-down* fashion, we are forced to implement *bottom-up* whole, self-consistent pieces of program before they can be tested by the interpreter. In the case of large programs this development approach (*think top-down, implement bottom-up*) [9] leads to a temporal shift between the specification and implementation phases thus limiting the advantages of using Prolog for writing executable specifications.

6. PROTALK: A FIRST STEP TOWARD SYMMETRY

In order to present the way ProTalk addresses the above shortcomings, let us introduce a little bit of terminology.
Given a Prolog program ***P*** we define the *meaning* of ***P***, ***M(P)***, as the set of ground unit goals deducible from ***P*** [9] (or, equivalently, the minimum Herbrand model for ***P***, [8]). In word, ***M(P)*** explicitly contains whatever ***P*** states implicitly.
We also define the *intended meaning* ***M***, as the set of ground unit goals that are intended to be true by the programmer.
We say that a program ***P*** is (partially) correct with respect to the intended meaning ***M*** if ***M(P)*** $\tilde{O}$ ***M***. We say that ***P*** is complete with respect to ***M*** if ***M(P)*** ***M***.
Let us denote with ***Pp*** the "partial program" as it is during the current refinement step; the development of a complete and correct program proceeds through a succession of partial solutions ***Pp***, incomplete and sometimes incorrect, that have to be enlarged and improved until a program P is reached such that ***M(P)*** $\int$***M*** (totally correct).
It is worth noting that ***M*** cannot be captured as a whole in the programmer's mind as being typically constituted by an infinite number of elements. Moreover the programmer is often unable to decide straightway if some ground unit goal belong to ***M*** just by looking at its form and at the problem analysis[7].
What the programmer can do is to recognize the logical relationships between some elements, or classes of elements, in ***M*** that he is sure of, and some other classes of elements (each one represented by a nonground unit goal) that should take part in the intended meaning. These logical relationships are expressed by Prolog clauses and constitute, at each step, the current ***Pp***.
Relying on the declarative semantics of Prolog, the programmer can only grant the *local* correctness of each clause unless he wishes to switch his mind to the "operational" mode and engage himself in a burden of error-prone computations.
What he needs is feedback about the overall meaning of the current partial program in order to verify its (partial) correctness, ***M(Pp)*** $\tilde{O}$ ***M***. The natural way for the programmer, to get this feedback, is to query the interpreter about those goals it guesses should belong to both M(P***p***) and ***M***, relying on its computational skill and on the equivalence of the computer-attributed meaning and the declarative meaning.
Unfortunately, when program development proceeds top-down, the most significant goals, the "top" ones, cannot be tested until we have firmly stated the lower goals they depend on.
ProTalk implements an extension of the idea of QtU proposed in [10], providing the programmer with an insightful feedback about the current partial program, whatever the development approach.
This feedback is realized by means of enhanced interaction with the programmer that accomplishes a common *pragmatic model* of communication, as required by the property c).
The most relevant point in such a model consists in letting ProTalk ask the programmer a question in the very same way he asks ProTalk: through Prolog queries. In [11],Vannini refers to the way ProTalk queries the programmer with the term EQtU, acronym for Extended Query the User.

Besides extending the interpreter's communication capability that of the programmer has also been extended, towards the definition of a full set of couples of complementary communication acts shared by both the agents.
Let us exemplify a typical interaction pattern with ProTalk pointing out the structure of nested couples of communication acts. We will refer to the programmer (Human) and ProTalk, with the labels *H* and *PT*, respectively, and with (open) and (close) to the corresponding opening and closure act of each couple. After *H* has provided a set of clauses ***Pp*** that should cover a goal ***Q***, he verifies his guess by querying *PT:*

H(open) : ?- ***Q***

As long as *PT* knows enough to fully expand ***Q***, it will act like any other Prolog interpreter. But, if it recognizes that it lacks some clause to expand a given subgoal ***Q'*** (which ***Q*** depends on) it suspends ***Q'*** and, before backtracking, asks through an EQtU if *H* can expand ***Q'*** :

PT(open) : can you expand ***Q'*** *-?*

PT relies on ***M*** as a model for the completion of ***Pp***.

In this way it tries to take on *H* as an alternative way to solve a goal just relying on the programmer's capability of recognizing the *local* correctness of logical relationships that might relate ***Q'*** with some other elements in ***M*** .
At this point *H* can open a further nested couple (if it is not persuaded of the need to expand ***Q'*** in order to solve the original goal ***Q***), asking *PT* to provide a "why explanation" about the last EQtU:

H(open) : why ?

Pt will replay showing the path, on the current proof tree, from the goal being subject to EQtU (***Q'***) toward the original root goal (***Q***).

*PT(close) : **Q'** to_prove **Some_intermediate_Subgoal** to_prove **Q***

In this way *PT* retrospects, in a forward chain fashion, the clauses it has used in attempting to solve ***Q*** , letting *H* verify if some of them are ill-defined.
As another choice, *H* can suspend the goal ***Q'*** and then ask for a new query to get further advice on the actual meaning of ***Pp*** , (returning later on ***Q'***).
Otherwise *H* can close the nested couple either letting ***Q'*** fail and, thus, allowing *PT* to backtrack:

H(close) : fail

or providing some clauses (facts or rules) useful to solve ***Q'*** , and then letting *PT* to resume its evaluation (taking into account the last provided knowledge):

*H(close) : add : **Q' :- Body.***

The above sampling should be enough to get a feeling about what interaction with ProTalk is , and how it could evolve.
All the relevant knowledge exchanged between the partners is conveyed using basically Prolog clauses: queries, facts and rules. Nevertheless, in the previous example, we can observe the occurrence of some non-Prolog statements such as the *add* and *why* command issued by the programmer or some accompanying natural language explanations , like *can you expand.*, issued by ProTalk. Furthermore, during a real session, on user request, help menu is presented, listing all the actions available in replay to the last opening act raised by ProTalk.
These features serve to support easy readability of dialogues from the user's point of view. Indeed, with some minor extensions (for instance, including built-in predicates for *why* and *how* explanations), the form of each Prolog communication act (query, fact, rule) and the context in which it occurs, should be enough for a correct understanding by both the agents.
The adoption of a "pseudo natural" logical language[8] might rule out any need to accompany commands or natural language explanations further enhancing the communication symmetry.
In a second implementation a new feature has been added to ProTalk for dealing with typical unfriendly behaviour of Prolog Systems: the cryptic *Stack Overflow* message that sometimes interrupts an evaluation. This extension called LQtU (for Loop QtU) enables ProTalk to prevent the occurrence of a possible infinite loop interrupting the evaluation and asking the programmer if there is something wrong or missing, or if he want to resume the evaluation. This kind of feedback is particularly suitable during the interactive definition of recursive programs in which some end condition is likely to be omitted.

The friendly interaction provided by ProTalk has been demonstrated to be valuable either in the extreme situation in which the programmer begins a working session asking queries with ***Pp*** empty. In this case the interpreter will reply with the request to specify all the subprograms needed to give an answer to that question, thus forcing the programmer to think in a strictly top-down fashion, about all different subproblems involved in the solution process. The feedback so provided is a helpful guide-line towards the problem decomposition task that must be undertaken in the early phases of program development.
The implementation of ProTalk is actually realized as a metainterpreter on top of Quintus™ Prolog which encodes the metaknowledge required for assuring the property c) of a symmetric communication.

7. CONCLUDING REMARKS

In our approaches, Prolog (with opportune extensions) accomplish two different communicational functions at the same time: interacting with the programming envi-

ronment, that is a typical concern of the programmer, and interacting with the program under development (as if it could be ready to deliver its active behaviour), that is mainly a concern of the end user. Here we can use the same language to encode knowledge and to interact with it. This remarkable attitude tends to blur the separation between end user and programmer, whose roles are traditionally kept far apart.

Going forward in this direction we are looking for a paradigm in which the same language of communication with a computer will serve, as a means of building up problem solutions and as a means to sharing results (with the involved knowledge) among several cooperating users. We think that symmetry could really provide useful insights toward this goal.

There are still several asymmetries in the communication with the prototypical environment implemented in Protalk and we are aware that it can not provide a general statement about the utility of symmetry in Man-Machine communication; certainly there are situations in which a less democratic paradigm could result more effective. Still we can look at symmetry as a powerful paradigm for exploring the area of computer mediated communications.

At the present several points are left open to verify : to what extent the paradigm can be applied to other programming environments either declarative or procedural; if it can be adapted to general purpose User Interfaces; if it can be supported by different media or form of utterance.

Acknowledgements

The idea of developing symmetry in man-machine communication in our department has been introduced and strongly supported by discussions with and the encouragement of Prof.G. Degli Antoni. Some students have enthusiastically developed their thesis around prototypical models of symmetry imbedded in PROLOG. These contributions come mainly from W. Vannini who experienced the EQTU first, from M.Castagnetti who extended and improved the implementation and from M.Terreri who was involved in applying the symmetry paradigm to the object-oriented programming, in PROLOG. To all of them, sincere thanks.

8. REFERENCES

[1] Orr W.D. (ed.), CONVERSATIONAL COMPUTERS, New York: Wiley, 1968

[2] Gaines B.R. and Shaw M.L.G, THE ART OF COMPUTER CONVERSATION, Englewood Cliffs, N.J.: Prentice-Hall, 1984.

[3] J.Weizenbaum, ELIZA - A COMPUTER PROGRAM FOR THE STUDY OF NATURAL LANGUAGE COMMUNICATION BETWEEN MAN AND MACHINE, CACM 9, 1966.

[4] G. Degli Antoni, MAN—MACHINE INTERFACE, nota interna, Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano, Milano 1984.

[5] Levinson, LA PRAGMATICA, Il Mulino, 85.

[6] W. Bibel, LOGICAL PROGRAM SYNTHESYS, Proceedings of the International Conference on Fifth Generation Computer Systems, Tokyo Oct. 1981, (ed.) T. Moto- oka, North-Holland Publishing Company.

[7] P.A. Subrahmanyam, THE SOFTWARE ENGINEERING OF EXPERT SYSTEMS: IS PROLOG APPROPRIATE? , IEEE Trans. Software Eng., No. 11, Nov. 1985.

[8] J.W.Lloyd, FOUNDATION OF LOGIC PROGRAMMING, Springer Verlag, New York, 1984.

[9] L.Sterling, E.Shapiro, THE ART OF PROLOG PROGRAMMING, The MIT Press, Cambridge MA,1986.

[10] M.Sergot, A QUERY—THE—USER FACILITY FOR LOGIC PROGRAMMING, New Horizons in Educational Computing, (Masoud Yazdani ed.).Ellis Horwood Series in A rtificial Intelligence, 1984, pp. 145—163.

[11]W.Vannini, UN INTERPRETE LOGICO INTERATTIVO PER L'ACQUISIZIONE DI CONOSCENZA, Tesi di Laurea in Scienze dell'Informazione, Università degli Studi di Milano, 1987.

[12] M.Castagnetti, PROTALK VERSO LA SIMMETRIA DELLA COMUNICAZIONE UOMO-MACCHINA, Tesi di Laurea in Scienze dell'Informazione, Università degli Studi di Milano, 1988.

[1] Here we refer to communicational roles regardless of the diverse skills, objectives and credit that each partner have in a specific task-domain

[2] Granularity refer to the amount of time and operations that occur between two subsequent communication acts.

[3] From a mere communicational point of view we shall not require that both agent attribute exactly the "same" meaning to each construct; nevertheless, in order to obtain a meaningfull communication, it is highly desirable that they can refer to a couple of somehow corresponding meanings with the same language's construct.

[4] Let us remark that the initial sharing of specific-domain knowledge is not a communicational requirement whereas it should be the very result of interaction.

[5] Some basic knowledge can be considered included in the background knowledge, likewise most programming languages encode some built-in functions.

[6] Let we say that this reasons should be enougth.

[7] For instance it is not easy to decide if *prime (15485863)* belong to the intended meaning of a primality-test program, even if the problem is well stated and a simple solution (perhaps not efficent) is easy to recognize.

[8] For a complete account on all the complementary couples of communication acts available in the actual ProTalk implementation and on their sintax, we remit to [12] and [13].

[9] The inclusion of imperative statements should be reduced as far as possible.

Questo articolo contiene errori di stampa gravi e fu ripubblicato corretto sul n. 51.

This article contains important typo errors and was fixed and republished on issue 51.

SELF-OBSERVATION IN ARTIFICIAL REALITY

Rita Pizzi
Dipartimento di Scienze della Informazione
Università degli Studi
Milano

RIASSUNTO

Poichè la Realtà Artificiale, in quanto rappresentata sullo schermo di un calcolatore, è a sua volta parte della realtà, ogniqualvolta tale schermo è compreso nello spazio di osservazione del soggetto si ottiene una riflessione infinita di livelli cognitivi. L'articolo formalizza tale situazione in un' "equazione della Realtà Artificiale", che viene dimostrato essere un'equazione al punto fisso. Questo risultato è confortato empiricamente da esperimenti sul videofeedback, dai quali emerge che la riflessione infinita dà luogo ad un'immagine di equilibrio che corrisponde ad un punto fisso - o ad altri attrattori - nello spazio delle immagini.
Vengono cioè generate strutture dissipative che implicano un aumento globale del contenuto informativo del sistema e sottolineano la natura temporale - oltre che spaziale - dell'autoosservazione.
L'autoosservazione quindi non è circolare perchè ogni riflessione si riferisce ad un tempo differente, che può osservare una realtà modificata.
Dunque un sistema in grado di autoosservarsi può trarre nuova informazione da questo procedimento, evitando situazioni di loop e potenziando le proprie capacità di problem solving.

ABSTRACT

As Artificial Reality, represented on the computer screen, is in its turn a part of reality, we obtain an infinite reflexion of cognitive levels whenever the screen is included in the subject's observation space.
The paper formalizes such a situation with an "equation of Artificial Reality", which can be shown to be a fixed-point equation.
This result is empirically supported by experiments on video-feedback, which show that infinite reflexion gives rise to an equilibrium image that corresponds to a fixed point - or other attractors - in the image space.
Therefore dissipative structures are generated, which imply an increase of the system's information content and stress the temporal nature of self-observation, apart from the spatial one.
Thus self-observation is not circular because every reflexion refers to a different time, therefore it can observe a different reality. A system able to observe itself can draw new information from this procedure, in such a way as to avoid loop situations and potentiate its problem solving capabilities.

The theoretical aspects of Artificial Reality make it possible to reconsider the deepest aspects of knowledge, of conceptual entities which constitute it, and their reciprocal connections.
In fact Artificial Reality reproduces on the computer screen in a meaningful way not only the reality image, but also its structural and cognitive connections [1].
Actually we note that Artificial Reality, represented on the screen, is in its turn a part of reality. Knowledge becomes this way a part of reality and as a part of reality it can be dealt with and elaborated.
Besides, we note that whenever the screen is included in the observation space, an infinite reflection takes place between cognitive levels.

The investigation of the concept of reflexive architecture can help to develop a realistic artificial model of intelligence. In fact intelligence can be defined as a system able to manipulate knowledge and to manage the reflexive interaction between different representation levels.
Many authors believe that interaction between representation levels and ability to manage this interaction are the key for the development of authentically intelligent systems [2].
In fact it is well-known that computational problems can arise when computers have to deal with complex recursive situations, like self-observation problems.
It is possible to show that such limits can be seen as computational implications of Goedel's Theorem.
The philosopher J.R. Lucas maintains that non-paradoxal self-observation takes place in human brain thanks to its ability to observe reality and its modifications, and to the following comparison with its internal computational version [3].
This way the brain can attribute truth values otherwise indefinite .
As a matter of fact, a conscious being can look into himself and examine his own actions without being different from the person who acts. On the other hand, a machine has to add new instructions to its program in order to examine it, and this way it becomes a different machine, which has to be examined in its turn, and so on.
The conscious mind can interrupt this circularity because meta-knowledge and knowledge are mixed into a single observer, and they are not an infinite tower of self-observers. It is a "fold-back", i.e. a feedback between different knowledge levels [4].
It is to be noted that an advanced computer program could perform this procedure, too: as the human brain can find out a propositional truth by comparing it with reality it perceives, as a computer can be endowed with a supervisor which compares the results of an internal subroutine with the external reality.
Therefore computational circularity can be avoided by endowing the system with the possibility of observing the modifications which occur in time in the reality it is part of.
We can analyze now these considerations according to the Artificial Reality approach.
Once we define an observation space on the screen - i.e. the portion of reality with whose artificial model the subject can interact -, we can put inside of the observation space a portion R of reality:

$$r(R)$$

A screen visualizes such a reality:

$$s(R)$$

Let us consider the situation where the screen s, which is depicting the reality R contained in the observation space, becomes itself inserted in the observation space. This modified reality may in its turn be visualized on the screen, and so on, originating the following infinite sequence

(r)		(s)
$r(R)$	$\longrightarrow$	$s(R)$
$r(s(R)+R)$	$\longrightarrow$	$s(s(R)+R)$
$r(s(R)+R)+R)$	.....	

(where we use the operation "+" between elements of R to represent space juxtaposition), so obtaining the recurrence equation

$$\mathbb{R}_{k+1} = s(\mathbb{R}_k + R)$$
$$\mathbb{R}_0 = R$$

If the resulting sequence admits limit, we denote it as

$$\mathbb{R}^{\wedge} = \lim \mathbb{R}_k$$

and we rewrite the equation of Artificial Reality [5]

$$\mathbb{R}^{\wedge} = s(\mathbb{R}^{\wedge}) + R$$

It is to be noted that the equation of AR is a fixed-point equation.

In fact, it is possible to consider the realities $\{\mathbb{R}_k\}$ as a set ordered by the relation of approximation c ,where information increases monotonically, i.e. the information content of $\mathbb{R}_k$ is included in that of $\mathbb{R}_{k+1}$:

$$\mathbb{R}_0 \quad c \quad \mathbb{R}_1 \quad c \quad ..... \quad c \quad \mathbb{R}_{k+1} \quad c \quad .....$$

because at each step a new screen is inserted in the observation space of reality:

step k ——> r (s (R) + ... + R)
step k+1 ——> r (s (R + ε) + ... + R) where ε = s(R).

The order relation generated this way satisfies the necessary conditions:

1) reflexivity:

$$\mathbb{R}_k \quad c \quad \mathbb{R}_k$$

2) antisymmetry:

$$\mathbb{R}_k \quad c \quad \mathbb{R}_h \quad c \quad \mathbb{R}_k \quad ===> \quad \mathbb{R}_{k=} \mathbb{R}_h$$

3) transitivity:

$$\mathbb{R}_k \quad c \quad \mathbb{R}_{k+1} \quad c \quad \mathbb{R}_{k+2} \quad ===> \quad \mathbb{R}_k \quad c \quad \mathbb{R}_{k+2}$$

Now it can be easily shown that $\{ \mathbb{R}_k\}$ is a complete lattice (6).
In fact,

- it is a chain, because Vk $\mathbb{R}_k$ c $\mathbb{R}_{k+1}$.
Anyway , it is easy to show that each pair $\mathbb{R}_k$, $\mathbb{R}_h$ of the set $\{ \mathbb{R}_k\}$ has both a meet and a join , consequently $\{ \mathbb{R}_k\}$ is a lattice.

- it is a complete lattice because it can be easily seen that every subset has a join.

c) if we put now

$$s'(\mathbb{R}_k) = s(\mathbb{R}_k) + R$$

we see that s' has the necessary features:

- monotonicity, because due to the information content we have

$$V \mathbb{R}_k, \mathbb{R}_h \ , \ \mathbb{R}_k \ c \ \mathbb{R}_h \ ==> \ s'(\mathbb{R}_k) \ c \ s'(\mathbb{R}_h)$$

- continuity, because it maintains the limits.
In fact, due to the lattice completeness, it is possible to write

$$s'(U \mathbb{R}_k) = U s'(\mathbb{R}_k)$$

where $U \mathbb{R}_k$ is the limit of the sequence $\mathbb{R}_k$.
It is in fact

$$s'(\mathbb{R}_k) = s(\mathbb{R}_k) + R = \mathbb{R}_{k+1}$$

and it follows that

$$s'(U \mathbb{R}_k) = U s'(\mathbb{R}_h) = U s'(\mathbb{R}_k)$$

i.e. the information content of $s'(\mathbb{R}_k)$ increases with continuity with respect to the information content of $\mathbb{R}_k$.

The following fixed-point theorems are now easy to prove:

I) Due to the monotonicity of s', a unique minimal element $\mathbb{R}^\wedge$ exists such that

$$s'(\mathbb{R}^\wedge) = \mathbb{R}^\wedge$$

i.e.

$$s(\mathbb{R}^\wedge + R) = \mathbb{R}^\wedge$$

II) Due to the continuity of s', a unique minimal element $\mathbb{R}^\wedge$ exists such that

$$\mathbb{R}^\wedge = U s'^{n}( )$$

where

$$m'^{n}( ) = m'(m'(.....m'( )....)) =$$
$$=m (m (.....m ( ) +R).....+R)+R)$$

and is the least element of R:

c $\mathbb{R}_k$ Vk.

Here it is

= R.

We can now observe that this result has been empirically supported by interesting studies of J.P. Crutchfield [7].
Crutchfield developed a system where a camera placed in front of a monitor converts the optical image on the monitor into an electronic signal that is then converted by the monitor into an image on its screen. This image is then electronically converted and in its turn displayed on the monitor, and so on.
The author proposes a continuum model of this video-feedback similar to the reaction-diffusion equations of chemical dynamics and biological morphogenesis. Video-feedback proves to be a dissipative dynamical system.
In fact, if we consider a single state of the video-feedback (an entire image), the state is specified by the intensity function I(x), and it follows that

$$I_{n+1} = T(I_n)$$

where I_{n+1} consists of two parts: the old image and the incoming image from the monitor screen. If we compute even the spatial diffusion which contributes to the intensity at a point, due to a spatial coupling to neighboring pixels, we have

$$I_{n+1}(\underline{x}) = L\ I_n(\underline{x}) + s\ f\ I_n(b\ R\ \underline{x}) \quad (.)$$

where L is the decay factor of the old image, R is the possible image rotation, b the zoom factor, f c [0,1] the parameter f/stop, s c [+1,-1] the possible luminance inversion.

Crutchfield observes that Turing's equation for the evolution of the field $\underline{I}=(I_1,....,I_n)$ of concentration variables for biological morphogenesis

$$d\underline{I} / dt=\underline{F}(\underline{I}) + D^2\ \underline{I} \quad (..)$$

(where $\underline{F}$ is the "reaction" dynamics of the concentration variables and D the matrix of spatial coupling and diffusion rate of the concentration variables) is analogous to the continuous form of (.)

$$d\underline{I} / dt = L\ \underline{I}(\underline{x}) + s\ f\ \underline{I}(b\ R\ \underline{x}) + s^2\ \underline{I}(\underline{x})$$

where s is the matrix of the spatial diffusion rate.
Now, Turing showed that the equation (..) gives rise to spatial patterns that can oscillate.
Actually, the activated video-feedback gives rise to an equilibrium image which corresponds to a fixed point in the image space.
This image can be shown to be an attractor if we note that whenever we perturb the system slightly (for example by waving a finger between monitor and camera), the system gives rise to transients which are shortly replaced by the original fixed point.
Depending on the parameter settings (rotation, contrast, magnification, and so on) we always obtain attractors, in particular limit cycles or chaotic attractors.

In conclusion, the video-feedback behaviour emphasizes the temporal nature of self-observation, apart from the spatial one.
It can be asserted that reality in video-feedback is not an endless iteration of identical images, but it generates dissipative structures, which imply an entropy decrease, therefore an increase of the information content.
It is obvious that the information increase becomes really meaningful to the observer if the reality R visualized by the screen is not static but dynamic.
Thus the iteration included in the equation of Artificial Reality is not inconsistent because at each step it refers to modified realities.
Self-observation is not circular because every reflextion includes at each step a different time, thus it observes a different reality.
Such a conclusion agrees with the theories above mentioned, which maintain that computational circularity can be avoided if the system is able to observe its own modifications.
A system able to observe itself can continuously compare its present state with the past states and draw new

information from the comparison.
Such new information can be enough both to avoid loop situations, and to potentiate the problem solving capabilities of the system.

ACKNOWLEDGEMENTS

I would like to thank Prof. G. Degli Antoni for his fundamental contribution to this paper, Prof. G. Mauri for his valuable advice, and Dr. F. Ponti, who pointed out Crutchfield's work to me.

REFERENCES

[1] Degli Antoni G., ARTIFICIAL WORLDS, Proc. II Conference of Hypertext User Group, Milan 1988.

[2] Maes P., Nardi D., ed., METALEVEL ARCHITECTURES AND REFLECTION, North Holland 1988.

[3] Lucas J.R., MINDS, MACHINES AND GÖEDEL, Philosophy 36, 1961.

[4] Hofstadter D., GÖEDEL, ESCHER, BACH: AN ETERNAL GOLDEN BRAID, The Harvester Press, 1979.

[5] Pizzi R., Degli Antoni G., AN INTRODUCTORY MODEL OF ARTIFICIAL REALITY: RELATIONSHIP WITH SPENCER-BROWN'S CALCULUS OF INDICATIONS, Proc. First Conference on Cyberspace, Austin 1990, in press.

[6] Scott D., DATA TYPES AS LATTICES, MS Lecture, Amsterdam, June 1972.

[7] Crutchfield J.P.,SPACE-TIME DYNAMICS IN VIDEO-FEEDBACK, Physica 10 D, 229, 1984.

HYPERMILAN

Marina Bonomi, Marcella Logli
Dipartimento di Scienze dell'Informazione
Università degli Studi
Milano

RIASSUNTO

Con HyperMilan si sono pose le basi per lo sviluppo di un sistema informativo di largo impiego per le informazioni su Milano, le quali, seguendo la logica della multimedialità, si presentano in forma testuale, grafica, spaziale, topografica.
Il progetto, inserito nel paradigma della realtà artificiale e nell'ambito dei sistemi ipertestuali, permette una grande semplicità d'accesso alle informazioni sulla città, cercando altresì di contribuire allo sviluppo di sistemi cognitivi che utilizzino modelli iconici delle entità coinvolte direttamente come elementi centrali della comunicazione uomo-macchina.

ABSTRACT

HyperMilan is a multimedia informative system about the town of Milan. It is based on the hypertext paradigm and the "artificial reality" metaphor that makes it a new way of communicating in the field of urban informative systems.
The methodology used during the development of the project has involved the formalism of semantic networks. These fit well the description of a particular reality or subject, which could be implemented in a hypertext environment.
The idea of HyperMilan has grown up together with the goal of implementing an informative system and setting up an experimental electronic enciclopedia were a lot of information could easily be accessed by everyone who needs or who has to have relations with the town, in order to know public, artistic, business, legal, economical, cultural life aspects, reducing the access time.

1. WHAT IS IT

HyperMilan is a Multimedia Information System about the town of Milan. It is based on hypertext paradigm and the Artificial Reality metaphor that make it a new way of communicating in the field of urban informative systems.
It is implemented with Apple Hypercard and developed with Hypertalk and LPA Prolog language. The choice of these tecnologies is due to their flexibility on building creative hypermedia and hypertext projects.

2. AIMS

The idea of HyperMilan has grown together with the purpose of realizing an informative system and setting up an experimental electronic enciclopedia were a lot of information should easily be accessible by everyone who needs or who has to have relations with the town, in order to know public, artistic, business, legal, economical, cultural life aspects, reducing the access time.
An effort has been done to make the man-machine interface the more intuitive, immediate and universal as possible.
Another important aspect considered has been the re-

search on the topographical and cartographical knowledge of the town of Milan.
The traffic problem has been involved in the work to offer a tool that promotes a better use of public transport and that can become a model for other related subjects in the urban and viability field.
At the end it was thought to give the users the possibility to take themselves in a personal way stand inside HyperMilan reality.

3. PROJECT DESCRIPTION

3.1 Methodologies and Architecture

3.1.1 Semantic networks

At the beginning it was necessary to do an informative exploration and a general research on the various aspects of the town to get a panoramic view about the project's interest domain: Milan reality.

Therefore a town representation was studied, which was the as exhaustive as possible for the citizen who has to interact with the town life and its institutional structure. The need to reach a system knowledge, as part of a world, is equivalent of doing a structure analysis of the system itself. Following this aim the semantic networks descripting formalism was used because it fits in representing a large Data Base knowledge well.

A semantic network is an oriented graph made by nodes which represent classes of entities, concepts, objects, situations of the considered domain, and arcs between these nodes which represent the relations existing between nodes. An arc connected with a single node determines a property of that node. By means of this kind of formalism, it is possible to represent a taxonomic knowledge where each level of taxonomy is a node linked with an *is_a* relation, as regard the concept of hierarchy.
The semantic network of Milan defines blocks of reality which become the hypertextual nodes of the project. In fact an analogy between semantic networks and hypertext does exist: hypertextual nodes can be imagined as representations of single ideas or concepts, links between nodes can be imagined as representations of semantic interdipendences between these ideas.

In this way the manifold aspects ("informative worlds") which build up the city life have been loomed and placed, with the respective interactions, in the whole context, including how the government system interacts with these aspects, so that the common citizen could navigate and interact whith a transparent reality of the city.
In fig.1 it is showed the scheme of the global semantic network which has been the starting point of the entire project.
In this picture an italian terminology is used to describe the political and administrative life as regards an Italian town.

3.1.2 Hypertext

An hypertext can be built to represent a semantic network. Hypertext nodes become single ideas or concepts and links become conceptual interdependences between nodes as said before.
All the project architecture is based on an hypertext structure. Let's see the hypertextual structure used in HyperMilan. Above all the hypertext is the instrument which lets support and manage different information elements and multimedia environments, as a navigation by users.
The informative domain of HyperMilan is the town considered in its wholeness, all the aspects are touched and there is no particular point of view. The user to which the project is oriented is either the citizen or the visitor.

The hypertext structure is organized in two parts: some hypertext nodes are linked together following a hierarchical predefined logic and some others following a cognitive and conceptual logic association.
The links are organized in a way that allows one to get the same information through alternative paths.
The navigation criteria used is intended to be representative of the various types of search related to the different kind of users.

3.1.3 Artificial reality

It is possible to represent the knowledge through the Artificial Reality metaphor, that is a conceptual instrument which allows one to give a model of the reality in

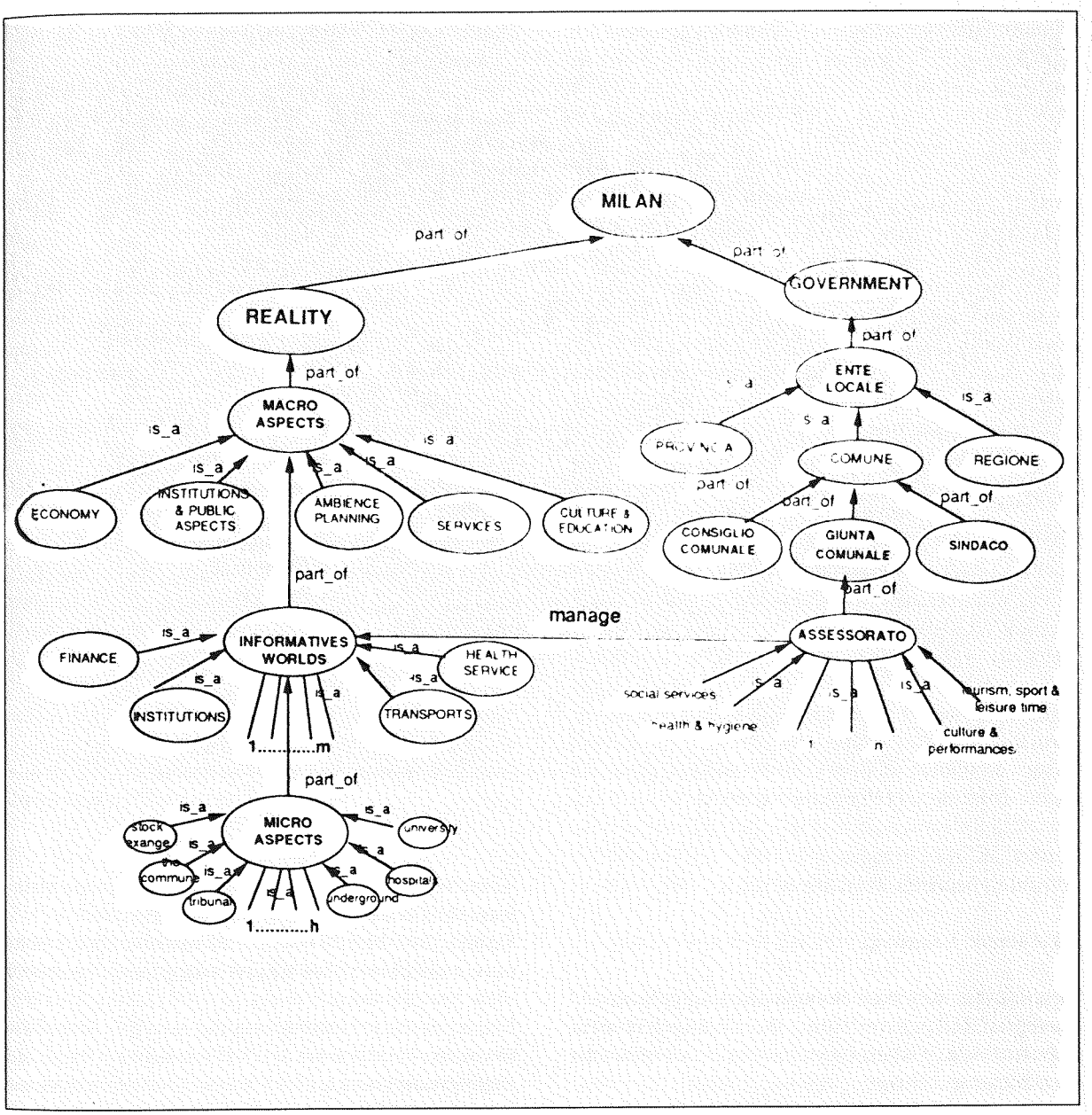


Figure 1: Semantic network on the general aspects of the city of Milan

a shape which is closer to the way it is perceived.
Till now anyone who wanted to know something had to choose between two alternatives: physically go to the involved place, or get information through other fonts (above all paper or computing ones). The Artificial Reality paradigm permits the integration of both these possibilities: the user moves in an Artificial Reality, reaching the desidered places (a street, a square, a monument, a museum, a building, the townhall and so on) which appear time by time on the screen becoming objects where he can get the required information.
In such a way the real town, made interactively a

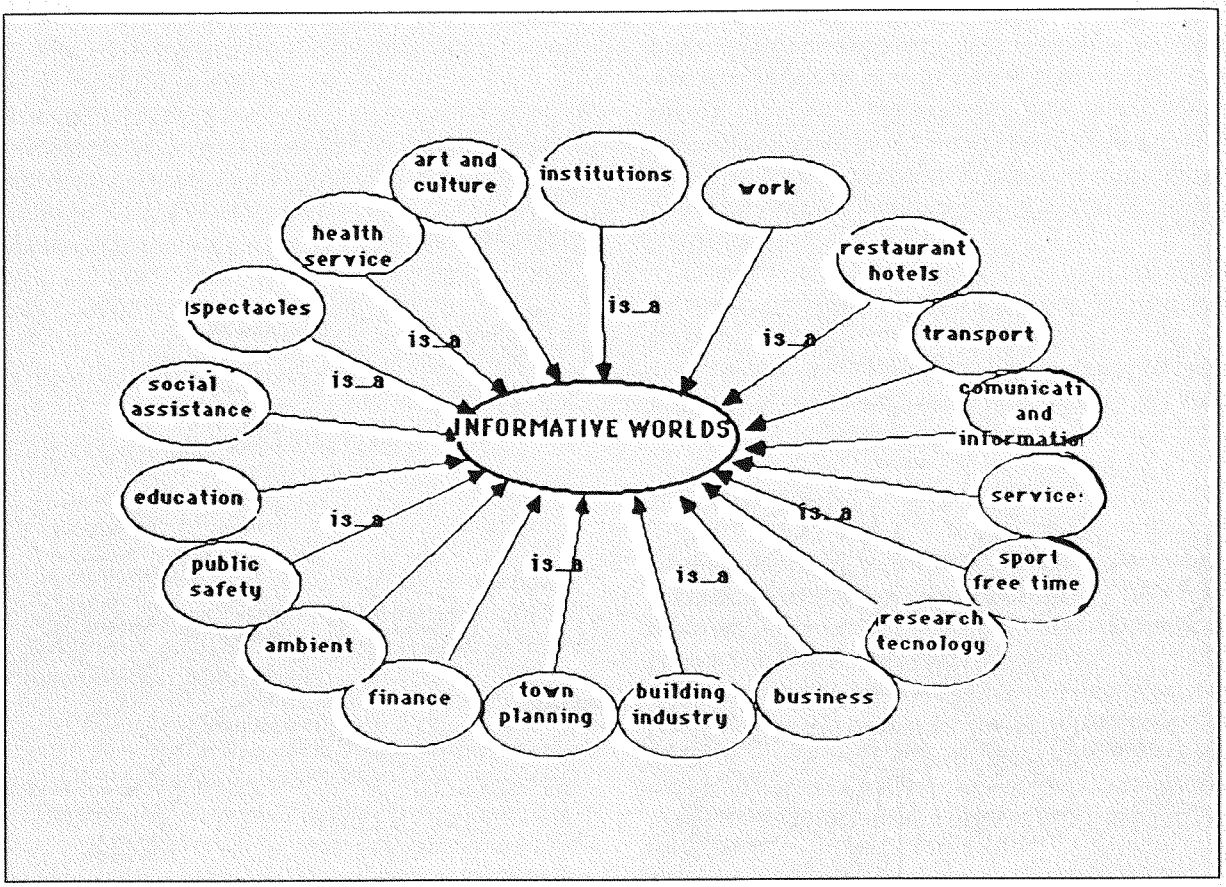


Figure 2: Istances of "informative worlds"

computer, becomes the artificial town.

3.1.4 Iconic Language

Describing a data base through icons, graphical and photographical pictures allows the user to perceive the town exactly as it is because of the immediateness of the comunicative message.
In this way each object in HyperMilan becomes a symbol of its significance.

3.1.5 Knowledge organization

To organize the HyperMilan knowledge meant to integrate all the concepts previously mentioned: this led us to represent the town with textual, visual and topografic information.

Therefore three environments have been created: an informative, a grafical and a cartographical one.
Besides three basical levels of informative abstractions have been pointed out related with citizens' points of view over the town.
The highest level corresponds to a general view of the town, the intermediate level is related to a local and more detailed area and the lowest level represents the most detailed information, both in pictures and textual information (fig.3).

3.2 Multimediality

HyperMilan has been organized so that the artificial town can be explored and consulted by the user with different approaches. Such different approaches are the ones usually referred to in daily life, according with the

different demands: for example "Tutto Città" catalogue is used to find a street (cartographic approach), the "Yellow Pages" can be consulted to know addresses and telephonic numbers informative approach), to visit monuments a person walks in the town streets (visual approach), and so on.
All these different approaches are available. This unfolds the creation of three ambiences:

1 - information
2 - the artificial town
3 - cartography

1 - Information

The "informative worlds" seen before in paragraph 3.1.1 are symbolically represented by the icons that appear in the main menu surrounding a schematic map of the town (fig.4).
The connections between the "informative worlds" and their accesses are made with buttons using words and pictures which users are familiar with.
Each icon button leads to a new menu concerning the icon subject itself.
Following the whole structure of the project, information is found starting by a more superficial level down to a deeper one, through menu buttons (fig. 5 and fig. 6), and viceversa, through an expansion button (the button down in the right corner of fig.6).

2 - The artificial town

The artificial town is the visual ambience whose main media is the photographical one. It is entered through the central highlighted area on the map (fig. 4).
The intermediate level is organized with city zone maps surrounded by icon buttons symbolizing some services available in that area. Through these maps the user can reach the desired place (a street, a monument place), that is the third level.
In fact the third level is characterized by digitalized images which visual reproduce the place (fig.7), flanked by icon buttons allowing an interaction with the image, overcoming films and photographs' limitations (fig. 8).
The user can follow some direction indicator buttons to move from a place to another of the artificial town, building his own path. Then, with the report button the user can get a list of the places he visited and the objects he encountered (monuments, shops, museums). This new method of browsing allows a kind of "history management", both regarding the user navigation and a way to communicate it.

3 - Cartography

A method has been developed to manage the maps in the computer in an interactive way.
Maps management is hypertextual and from a general map it is possible to go to a single zone enlargement (directly selecting it) constituted by toponymic maps. Here it is possible to get topographical information, for example to have the location of the streets on the map (fig.9). It is also possible to pass from a map to a contiguous one selecting a border area of the map. Some interface expedients have been studied to maintain the urban orientation.
Inside the cartographic environment a way to make maps thematic spaces has been developed: the aim is to give the locations of services or other interesting elements belonging to the same group (hospitals, gardens, transports, etc..) appearing on the general map in a whole vision.
For example, in the transport thematic map, it is significant to see the underground schematization that becomes an interactive object to make the user know how to use the tube choosing the best path to go from a stop to another one (fig.10). This function, in particularly, was developed in LPA Prolog language to solve the problem related to the best minimum path.

3.3 Interaction with the user

Concerning the interaction with the user one thing has to be underlined: in the informative ambience, there is a method for a dynamic information management that allows its updating and the personalization of it by a super user who could act with specific options. For istance he can make a dynamic link between pictures and information (that is to join together two different ambiences).
Therefore this means inserting peculiar information of a single object
(a shop, a gallery, an hotel, etc..) creating a special personal space.

	visual ambience	informative ambience	cartographic ambience
first level	aerial vision	general information	general vision
second level	closer aerial vision	detailed information	topographic vision
third level	ground vision	peculiar information	

Figure 3: Different environments in HyperMilan

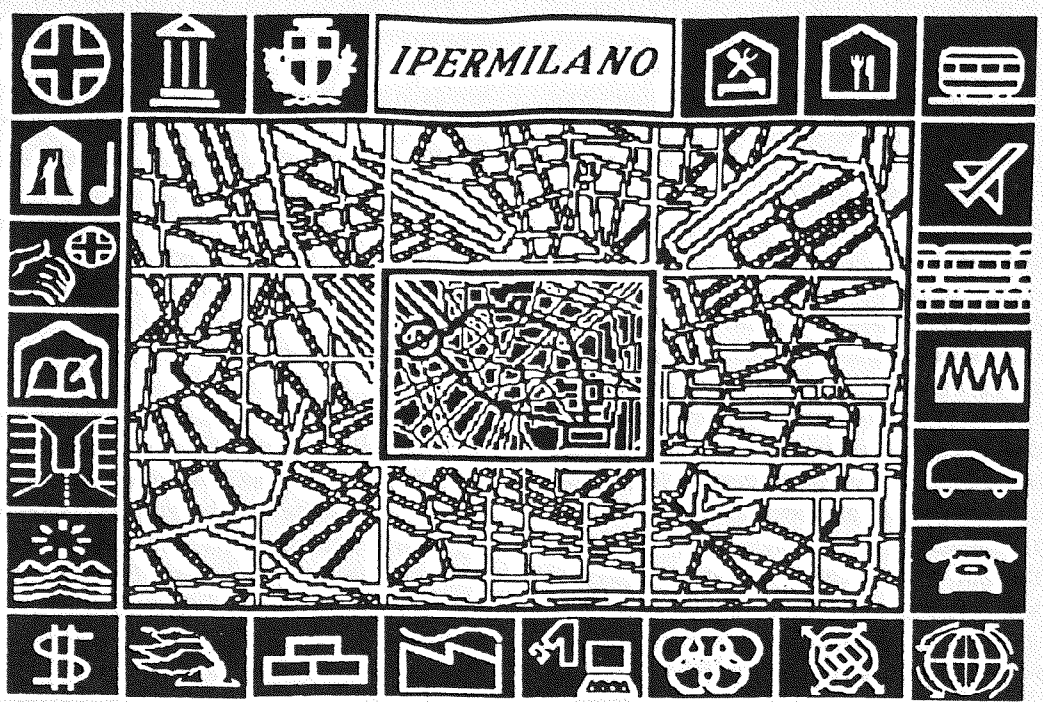


Figure 4: Menu card

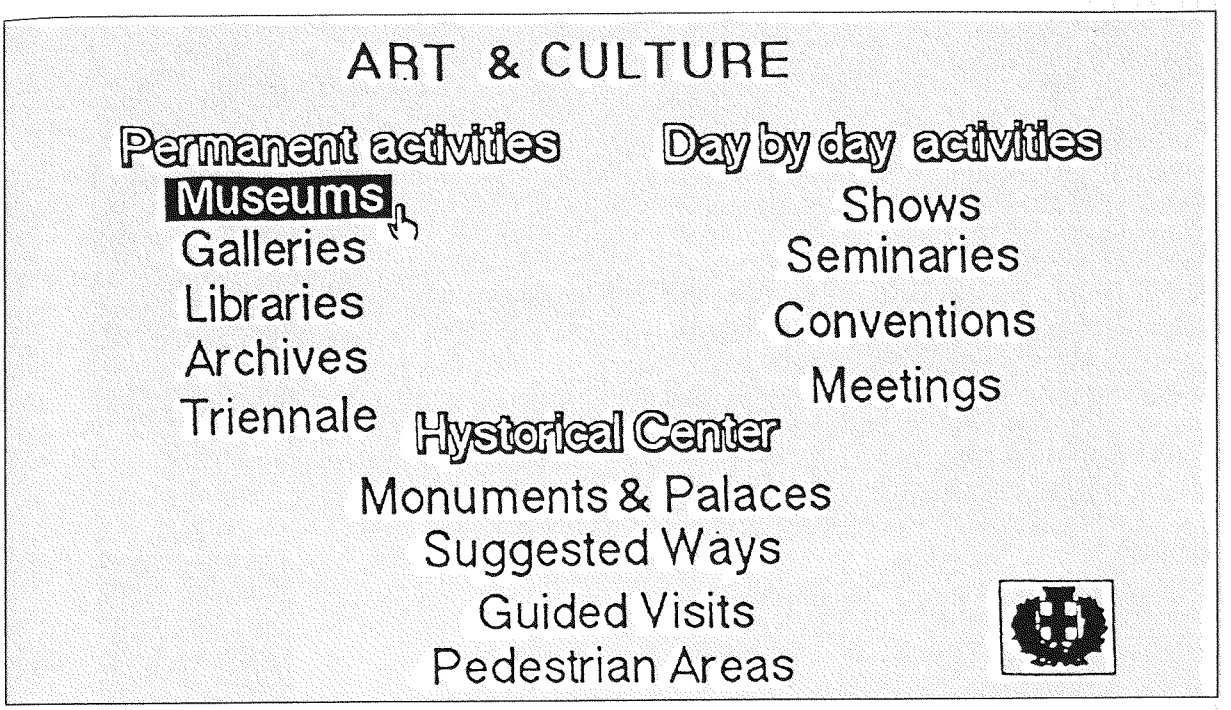


Figure 5: An example of "menu buttons"

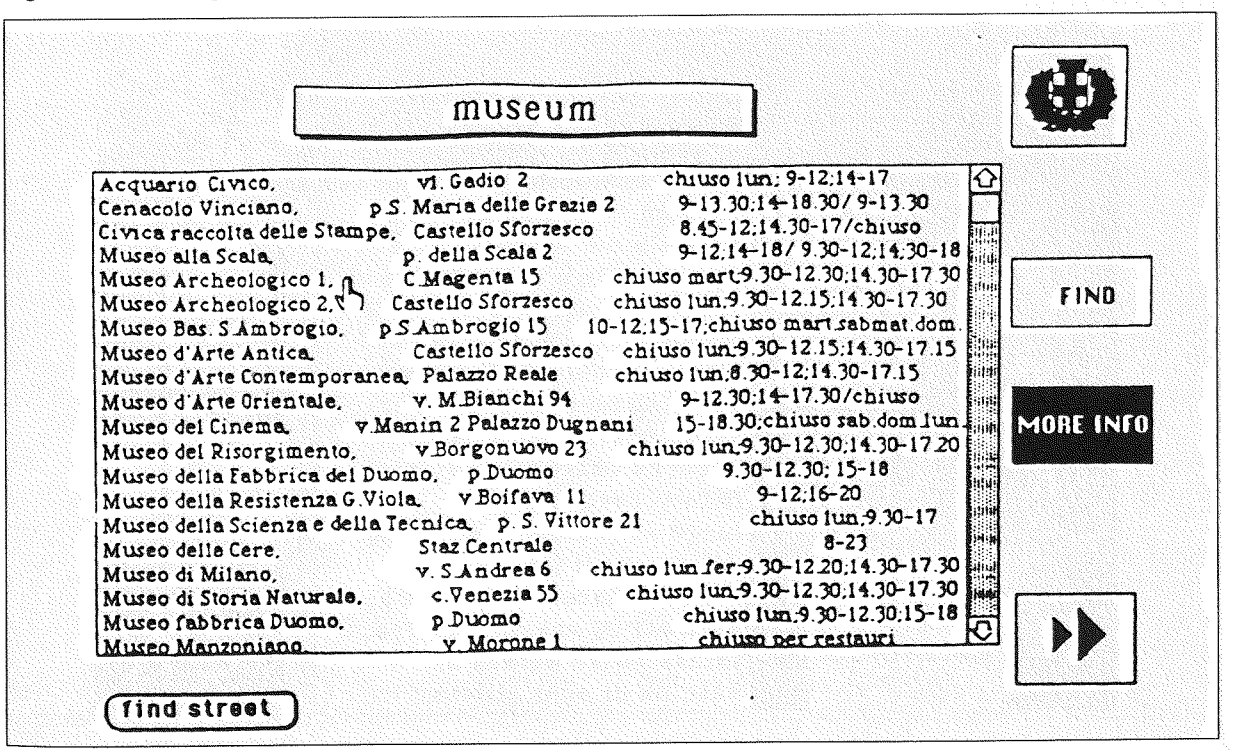


Figure 6: The general museums map

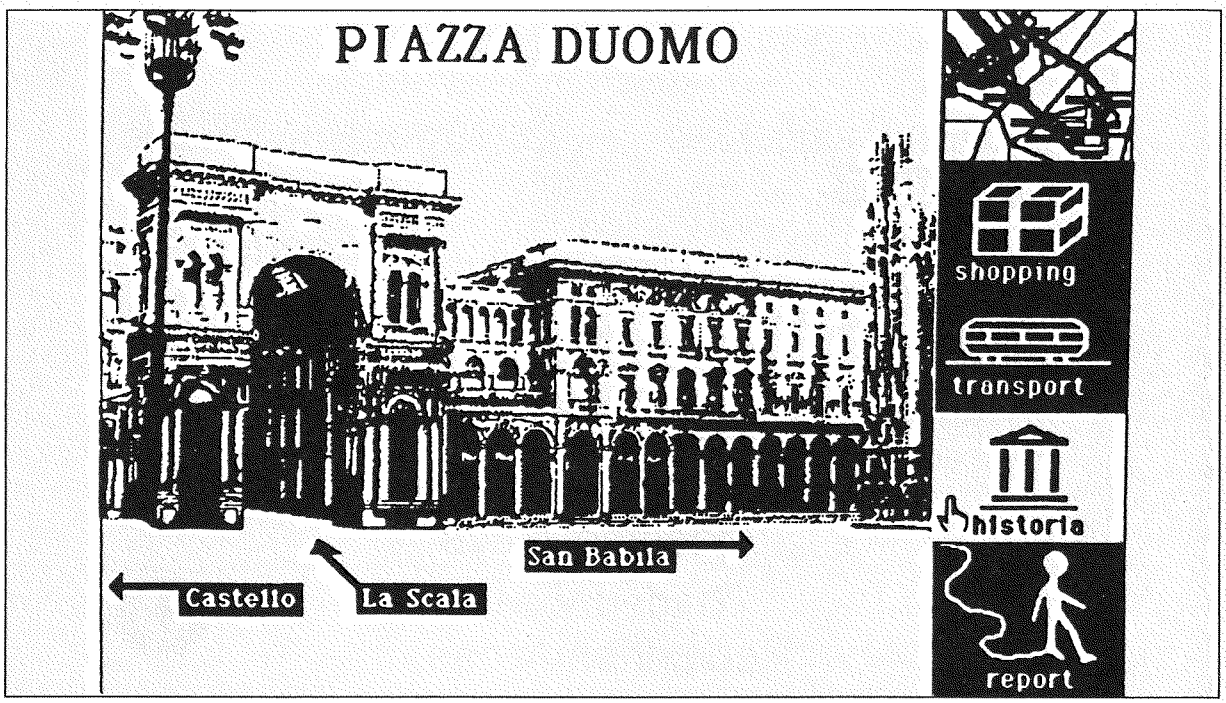


Figure 7: Navigating in the city

Figure 8: Clicking on the ''historia'' button and on a monument highlightedm the user can get hystorical information about that monument

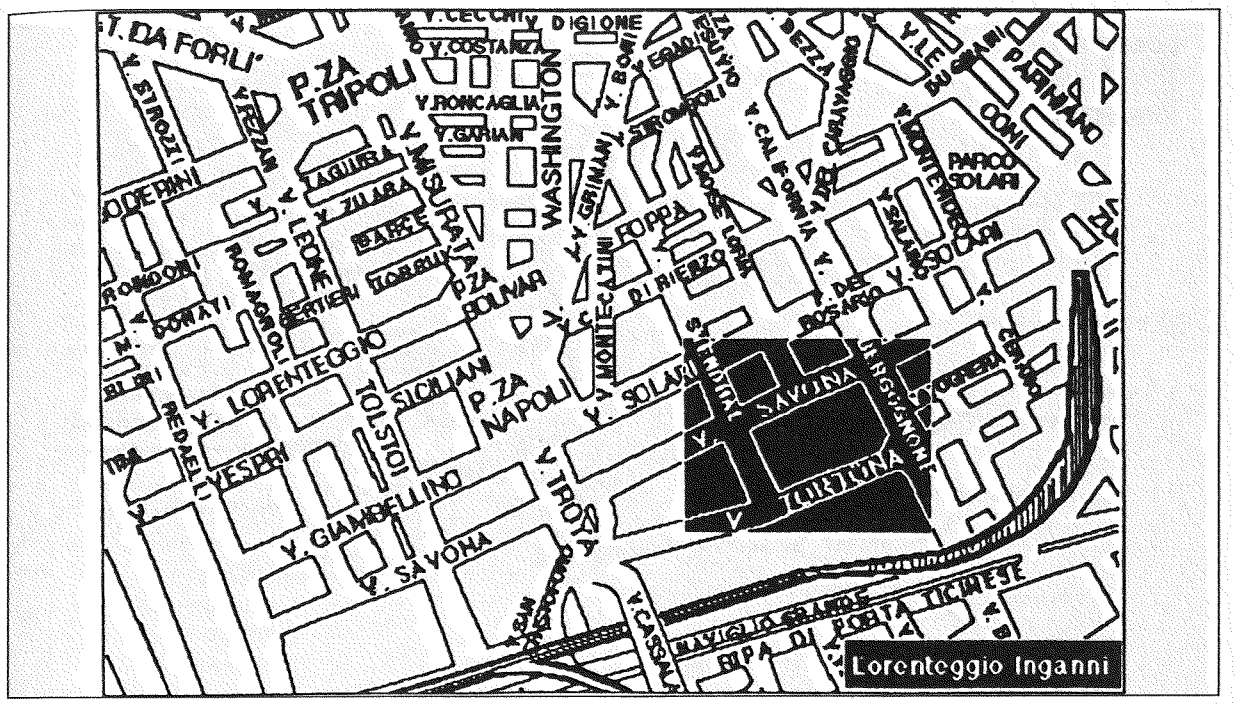


Figure 9: Topographic map

3.4 Navigation

Till now the project in its components and environments has been considered. Now let's see how they are related and so how the user can navigate in HyperMilan.
The navigation takes place both in vertical and horizontal directions: the former is the navigation through the three information levels, the latter is the possibility for the user to pass from an ambience to another one which is cognitively associated or to go along the same level of the same ambience.

Here is the network which shows how HyperMilan organizes its hypertextual links enabling the navigation into the artificial town seen as domain of knowledge (fig.11).

4. POSSIBLE APPLICATIONS

HyperMilan is a system for giving information, for providing deeper knowledge, and this is why it has numerous and various possible applications.
In fact the systems involves as many different type of questions as many different kind of users, because of the width of its informative field referred to every urban argument.
Waiting for such instruments being into everyone's house, a middle step is to find some useful places, where the citizens can meet them and enjoy them.
The places might be the following ones:

- Public places, that is interesting points spread in the town such as stations, aereoports, city centre, and so on.

- Public places as most attended as the previous ones, but less risked
(not being open air), such as town turist offices, museums, and so on.

- Institutional places, like Telegraph Offices, Municipal Offices, Consulates, Embassades, Health Service points, with the idea to create purpose information in order to improve the relationships between the citizens and the Institutions.

- Private places, that provide services for their clients and by the other hand advertize themselves: shops,

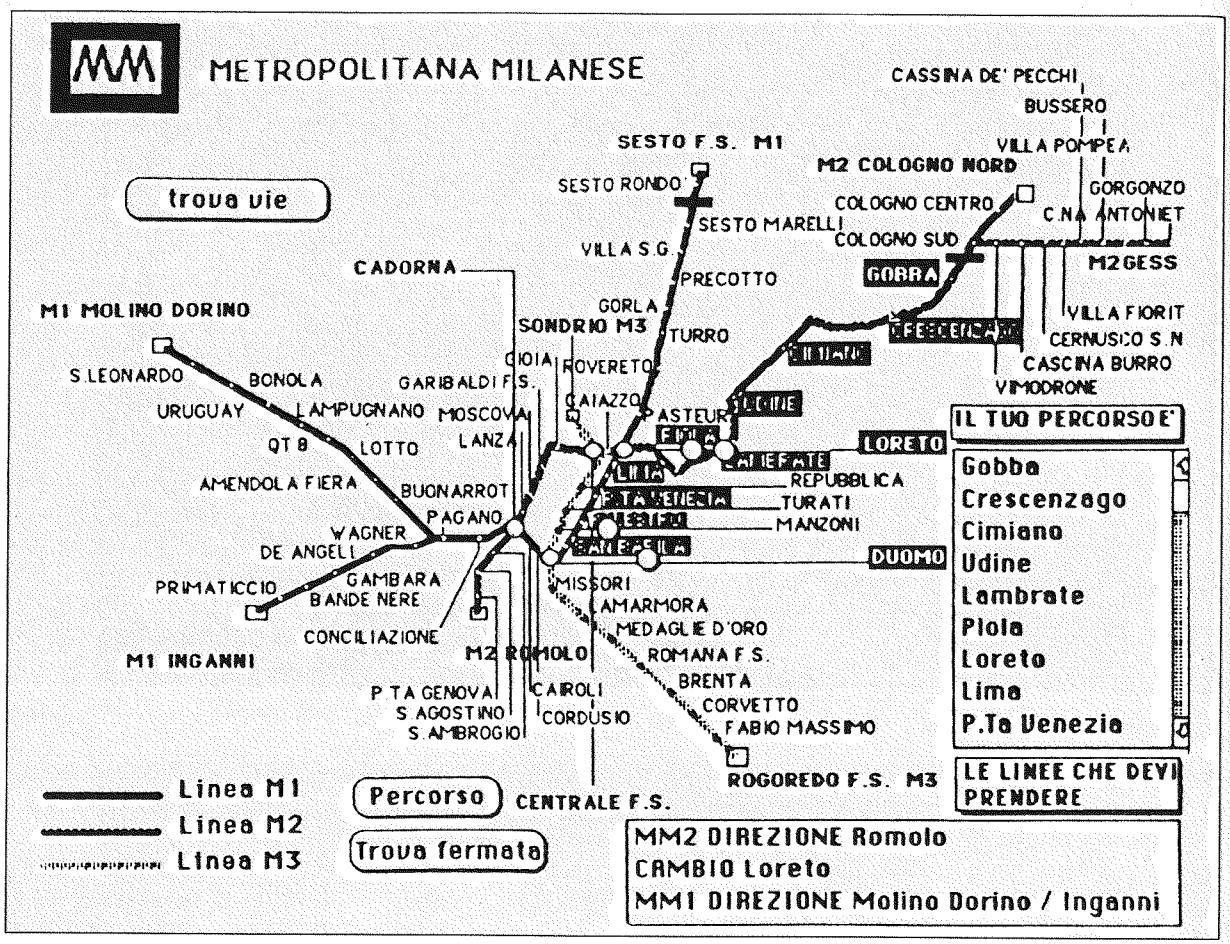


Figure 10: The underground map can be interactively queried

hotels, travel agencies, teatres, and so on.

- The system could be used for city appointments and events, shows, meetings, expositions, as a support and an integration, to the presentation and the organization.

- It could be very interesting and looking ahead, to suggest and support HyperMilan diffusion through an electronic publishing medium.
The updating of the time variable events would be published periodically.
In this way the system could reach everyone who has got a fitting computer: private houses, industries, professional offices, public and private organizations.

- Another kind of more specific and at the same time largely usable application could be giving a urban public transport information service to people moving in the town.
The topic of transports' has already been approaced starting with the underground. Now it could get a system that receving two entries, a starting city place and an arrival city place, gives back the public transport involved and their path through the town.
Such an instrument could be strongly supported by Administration, fitting the actual promotion policy of public transport use very well.

- With some purpose choices, the system could be managed to obtain a knowledge feed-back from the users' navigation paths, for studies and statistics.
In the same point of view, HyperMilan could also be

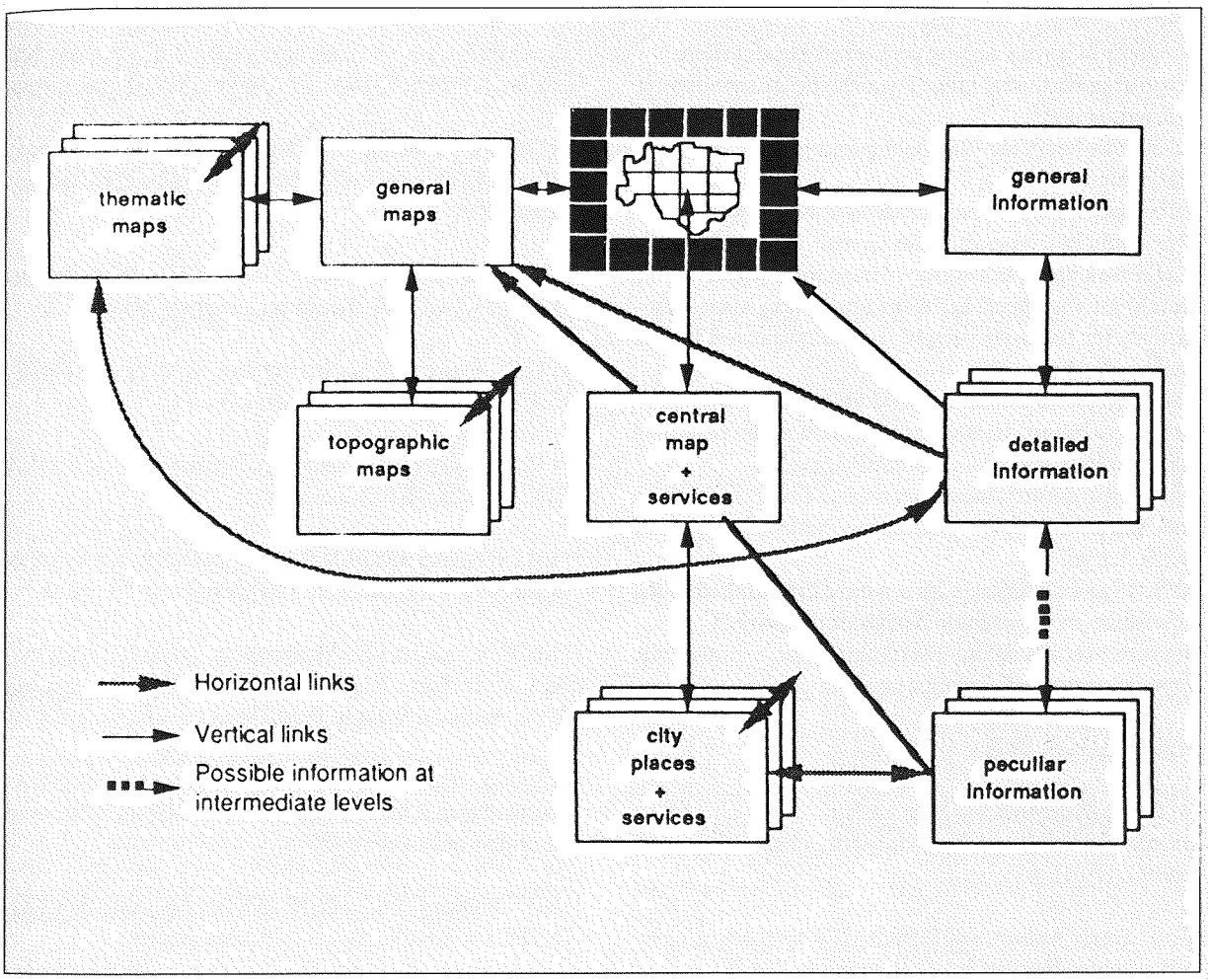


Figure 11: Scheme of HyperMilan hypertext

used in preparing some questions set about actual problems (the ambience, the pollution,and so on) to know what citizens think about. In this way it's possible to make the collaboration between citizens and Istitutions improve; therefore it would become possible for the town council to choose new political strategies, directly based upon the people's needs and behaviour.

Making an instrument like HyperMilan means being advanced in the studies about new communication media, and walking along the development way, which in a few years shall lead to obtain, in an international scale and in a familiar diffusion, the add-value services, that are the tomorrow's HyperCities.

5. CONCLUSIONS

The problems met involved a data base management system more suitable to such a huge amount of urban information as a metropolis rather than one where it is possible to build with the Hypercard tools.
For the great importance and presence of maps and pictures in HyperMilan, some difficulties arose in reaching the images high precision degree expected.
The transport model developed in Prolog has risen many questions regarding the minimum path problem solving and the communication between Prolog and Hypercard environments.
With new technologies (Supercard and Plus) it could be

possibile to have a colour version of HyperMilan.
Some arguments that make HyperMilan an innnovative instrument are:
• the idea of appling hypertext paradigm to a metropolitan space
• the topographic research and its management
• the study of transport subjects with A.I. methods
• the developing philosophy of an iconic, cognitive and metalinguistic interface for the navigation in the information, for the carthographic consultation and for the choice of urban paths
• the navigation in the artificial town
• the possibility of becoming a part of HyperMilan reality with a personal informative space.
The project is the Graduation thesis of Marina Bonomi and Marcella Logli, guided by Professor Giovanni Degli Antoni.
Roberto and Gualtiero Carraro have contributed to the realization of the graphic design of the project.
Roberta Macchi, HUG coordinator, has followed the developement of the work entirely.

6. BIBLIOGRAPHY

[1] IRER (Regional Institute for resarch of Lombardia) Giorgio Pastori (resarch coordinator), PROGETTO MILANO, 1983_87

[2] G. Degli Antoni, ARTIFICIAL WORLDS, Conference proceedings of HUG II meeting, 1988h

[3] G. Degli Antoni, FROM REALITY TO HYPERMEDIA TROUGHT ARTIFICIAL REALITY, DSI internal note, 1988

[4] G. Degli Antoni, IL COMPUTER, IL REALE, L'ARTIFICIALE, Note di software n.41, 1988

[5] G. Degli Antoni, MAN MACHINE INTERFACE, DSI internal note, 1988

[6] G. Degli Antoni, IL COMUNE INTELLIGENTE, Note di software, n.41, 1988

[7] Conference proceedings HUG: II MEETING, december 1988

[8] Conference proceedings VERSO LA COMUNICAZIONE ELETTRONICA, Il Sole 24Ore, April 1989

[9] Malpas, PROLOG: A RELATIONAL LANGUAGE AND ITS APPLICATIONS, Prentice Hall International 1987

[10] Goodman D., THE COMPLETE HYPERCARD'S HANDBOOK, Bantam Computer Books

[11] K. Weiskamp N. Shammans, MASTERING HYPERTALK, Bantam Computer Books

[12] G. Caja, P. Piardi, C. Testero, RASSEGNA SUGLI IPERTESTI, june 1988

[13] J. Conklin, HYPERTEXT: AN INTRODUCTION AND SURVEY, J.E.E.E. Computer vol. 20, N. 9

[14] E. Charniak, D. McDermott, INTRODUZIONE ALLA INTELLIGENZA ARTIFICIALE, Masson SPA- Addison Wesley Publishing Company, 1988

[15] G. M.R. Nilsson N.J., LOGICAL FOUNDATION OF ARTIFICIAL INTELLIGENCE, Morgan Kaufman Publisher 1987

[16] Nilsson N.J., PRINCIPLES OF ARTIFICIAL INTELLIGENCE, Tioga Publishing 1980

[17] W. Bern R. Grahm, IL PROBLEMA DELLA RETE DELLA LUNGHEZZA MINIMA, Le Scienze ediz. italiana di Scientific American, marzo 1989

Special thanks to Professor Giovanni Degli Antoni for his guide in the whole project, to the Hypertext User Group, particulary to its director Roberta Macchi, for the collaboration, to Roberto Grande for his advice and to Roberto and Gualtiero Carraro for their artistic contribute.

THE CONCEPT OF REUSABILITY IN EDUCATIONAL DESIGN

Giorgio Olimpo
DIST, Università degli Studi
ed
Istituto per le Tecnologie Didattiche del CNR
Genova

RIASSUNTO

L'articolo illustra alcune problematiche relative al concetto di riusabilità per il software didattico multimediale. Viene tracciato un parallelo con quanto avviene in altri settori ed in particolare con quello dell'ingegneria del software e succesivamente vengono discusse le principali classi di requisiti per l'effetiva riusabilità dei componenti multimediali (proprietà logiche e didattiche, accessibilità, modificabilità, interoperabilità, proprietà di gruppo).

ABSTRACT

The concept of reusability, which is already well established in several branches of engineering, is still little understood and practised in the field of educational software. The paper makes an attempt to model the concept of reusability in the educational field by describing the basic requirements for the actual reusability of multimedia educational components (logical and educational properties, accessibility, interoperability, modificability and group properties).

1. THE NEED FOR REUSABILITY IN MULTIMEDIA COURSEWARE

The amount of effort which goes into producing high quality multimedia material is usually very large and the level of skills required very high. This is the reason why it is extremely important for the future of the field of the application of multimedia interactive technology to education to find efficient ways to share the efforts of couseware development. One very appealing approach is that of reusability. The word reusability evokes the idea of adapting and combining already existing *reusable* components of learning material instead of starting every new authoring effort from scratch.
The reasons for reusability may be summarized as follows:
- reducing the development cost
- improving the quality
- being able to give quick answers to educational needs.
In fact the concept of reusability goes hand in hand with the concept of rapid prototyping.

1.1 Reusability in other fields

The concept of reusability is widely known and practised in several fields of engineering (even if often they do not use the term reusability). Civil, mechanical and electronic engineering have a long tradition of designing by using existing and commercially available components. Catalogues of components are commonly available. Sometimes these components may be very complex as in the case of an integrated circuit in electronic engineering.
In the field of software engineering, reusability is a

more recent phenomenon. The first attempt of reusability was based on functional abstraction and the first practical tools to implement this concept were libraries of subroutines. The next step was data abstraction which supported a variety of linguistic tools (classes, clusters, modules, packages,...). The most recent achievement has been object-oriented programming which, by enhancing data abstraction with inheritance, provides a sound mechanism for supporting reusability.
In the field of education, reusability is little more than a dream. It is true that there are already some examples of reusing educational material. A typical case is the *repurposing* of existing videodiscs, i.e. adapting a videodisc to meet an objective which is different from the original by building a new delivery software. However cases such as these implement only a very restricted concept of reusability as it will become clearer below. It is true that in building the concept of reusability in education several ideas may be borrowed from the field of software development and from object-oriented programming. However, the specific characteristics of the fields of multimedia and education require that, to some extent, the concept of reusability be remodelled. Major problems arise from the difficulty of building decontextualized learning components and the interoperability of different, possibly very complex, components.

1.2 Decontextualization of learning components

In this area it is very easy to make comparisons with the field of software development. Classes such as *collection* or *linked list* can be thought of as objects which are reasonably independent from any specific application and it is highly likely that they will be reused in many different contexts. Unfortunately no equivalent situation can be found in the field of education where the degree of complexity is much higher. In education, most concepts are not completely understood and formalized; the concept of correctness is much more blurred than in the field of software development; and so far no equivalent for the concept of class, which is the basis of the object-oriented approach, as been found. Not only do all these factors make it almost impossible to define context-independent components but, even within the same learning domain, it is very difficult to define components which can be easily reused by different applications.

1.3 Interoperability

One of the key aspects of reusability is the design of new applications by linking together existing components or applications which, in most cases, were not originally built for working together and, in some cases, were even designed to work as independent applications.

2. REQUIREMENTS FOR REUSABILITY IN EDUCATIONAL DESIGN

Some of the obstacles to reusability are at least partially of a psychological nature. "The *not invented here* syndrome is well known. The implication of this in practice is that reusable components should bring a significant advantage over home-brewed ones in terms of quality, ease of access and cost. A marginal advantage will not be enough to convince authors to use somebody else's mousetrap rather than invent their own." (*adapted from Bertrand Meyer).*
The following are some of the requirements for the actual advantageous reusability of learning components:
- Logical and pedagogical properties
- Accessibility
- Modificability
- Interoperability
- Database properties
- Multilinguality

These properties are briefly discussed below.

3. LOGICAL AND PEDAGOGICAL FEATURES OF A REUSABLE BLOCK OF LEARNING MATERIAL

Following are some of the logical and pedagogical properties of a reusable multimedia component. The list is not intended to be exhaustive.
a. usability
To be reusable, a block must first be usable, not perfect but usable. This means that a component should exhibit properties like understandability, correctness, quality, etc.
b. coherence
The presence of different conceptual, independent or loosely coupled aspects within the same component makes the component itself difficult to use without modification, and difficult for the user to manage from a

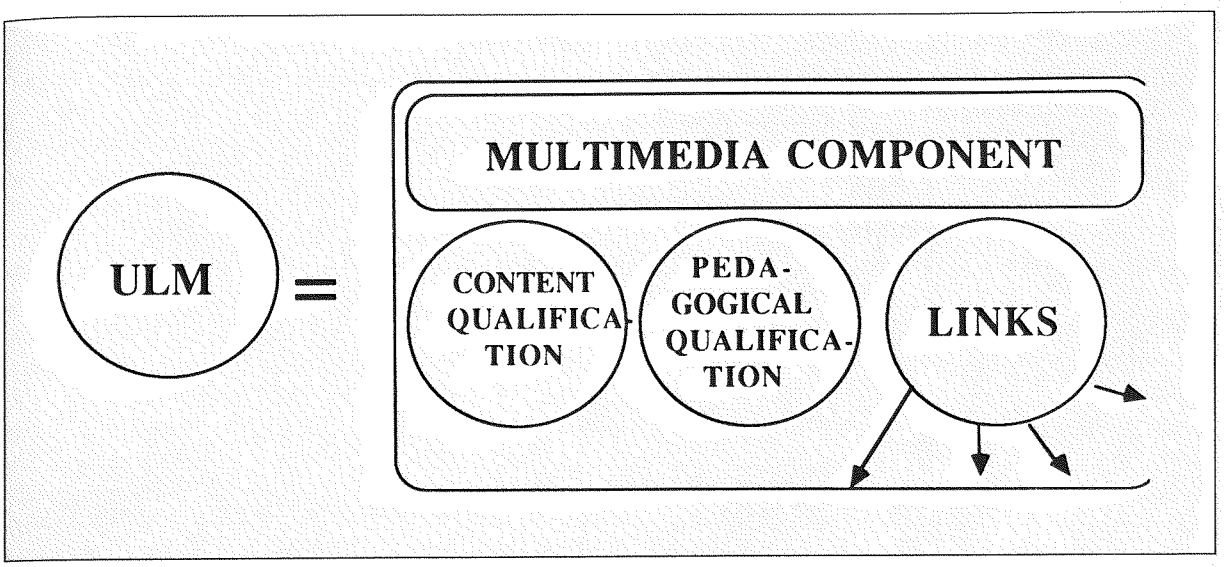


Figure 1: The structure of a Unit of Learning Material (ULM)

logical point of view.

c. completeness

Each block should exhibit a unique communicative, informative or educational function (ranging from very simple to very complex) and perform it completely. The completeness of the *educational* function allows the author to use a block without needing to retrieve another complementary block.

4. ACCESSIBILITY

"The best reusable components in the world are useless if nobody knows they exist, if it takes a long time to obtain them or if they cost too much. The practical success of reusability techniques requires the development of an adequate database of software components, which may be searched by appropriate keywords so that a potential user will find out quickly whether some existing component satisfies a particular need"(*Bertrand Meyer)*

As a further enhancement, "network services should also be available, allowing electronic ordering and immediate down loading of selected components". These statements, which refer to the field of software development apply equally well to the field of courseware development. However, in the later case, it is important to make some refinements. In fact, in the field of educational design, a traditional query mechanism might not be sufficient to cope with the needs of the authoring process and might need to be integrated with a navigation mechanism. For example, when a component X has been retrieved by an author via a query on the database, it may be very helpful to reach the next useful component Y by navigating a link network. This is especially useful if it is highly likely that Y is directly linked, or at least very close, to X. In fact, in this case, even if the author does not exactly know what he/she wants, through navigation he/she may reach something that is helpful. In other words, to build a really efficient mechanism of access, we have to merge the concepts of database and hypermedia. "The information in the database should be heavily cross-referenced by links, the majority of which should be created and maintained by the system, rather than being explicitly maintained manually. This automatic linkage-creating mechanism will be crucial in producing databases of highly reusable material." (*Maurer*).

Fig. 1 shows, in principle, the structure of a Unit of Multimedia Learning Material (ULM). A ULM should include, beside the physical description of a multimedia component, a set of links to support navigation and sets of attributes to support queries. It should be noted that one of the sets of attributes refers to the pedagogical qualification of a multimedia component, i. e. the description of its pedagogical features. This set of attributes may be very important in supporting both the autho-

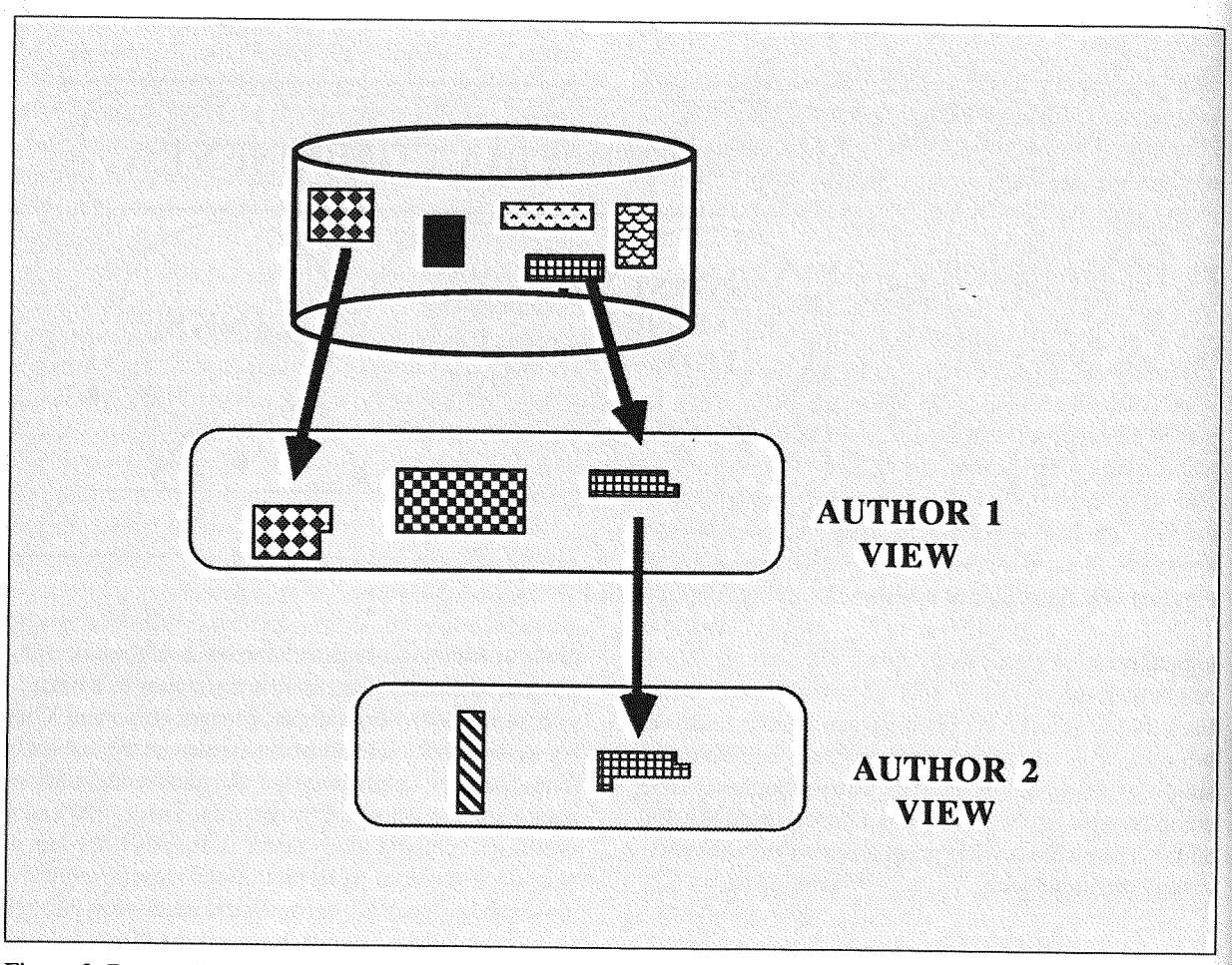


Figure 2: Personalized views of the same database

ring process (an author may be looking for a particular component with given educational features) and learning processes based on guided navigation (an intelligent system, for guiding the learner towards a specific educational goal, may need a component with given educational features). The definition of the educational features characterizing a component is critical and much experimental work is needed to assess the existing proposals. Some examples of pedagogical attributes are educational function, learning objective type, knowledge type and approach type. Some of these attributes, such as learning objective type, are characterized by a set of values which are domain independent while others, such as approach type, have a set of values which are partially dependent on the teaching domain. It should be noted that, in some cases, widely tested taxonomies are already available, as in the case of learning objectives for example, there are many taxonomies, of which Bloom's is certainly the most well-known. For other attributes such as educational function or knowledge type the definition of similar taxonomies requires further research.

5. MODIFICABILITY

Each author or user of the database should be able to modify existing blocks to adapt them to his/her actual needs. This could simply be considered as an editing problem, easily dealt with by a good multimedia editor. However, we believe that another factor, which deeply affects the structure of a database of reusable blocks, should be mentioned. At any given moment, a user may

want to personalize some of the blocks in a DBLM by adapting them to his/her needs, and then later refer to this personal version of the database, which now consists of the original version enhanced by the modifications he/she has made. Since a database can be accessed by different authors having different personalization needs, some provision has to be made to support personalized views of the same database. The most natural solution seems to be to organize a database as a hierarchical structure endowed with some kind of inheritance mechanism (see fig. 2). In object-oriented programming the inheritance mechanism provides the basis for personalization. Fig. 2 illustrates a mechanism which is rather different from the class inheritance typical of object-oriented programming. Here each personalized view inherits the knowledge of the reusable components in the parent view. However, the user has the right to modify some of them or add new ones to them.

6. INTEROPERABILITY

This property refers to the possibility of exchanging multimedia material between different hardware/software environments and the possibility of linking components or applications which were not originally built to live together.
The solution to the first problem is dependent on document interchange standards for multimedia documents. At the moment, comprehensive proposals are still to come. ODA/ODIF, POSTSCRIPT, RTF (Rich Text Format), VEX (Video Extension for X-Window) are all existing, de facto or proposed standards which are mostly oriented to textual or graphic material or, as in the case of VEX, try to extend existing standards to multimedia documents. An ISO commission is presently working on this problem which is crucial to the process of sharing multimedia material.
The second problem is more complex since, at the moment, it is not realistic to think of standards or a uniform discipline accepted by application developers (which would also exclude most of the existing applications). Therefore specific research is needed to define protocols and tools for interoperability.

7. PROPERTIES OF A DATABASE OF REUSABLE BLOCKS

A user should be able to find in a pool or database of reusable components a significant portion (hopefully the majority) of the material he/she actually needs. A DBLM should not be only a marginal help, but a substantial help to the work of the author. A database may be oriented either to a specific content domain and to several classes of users or to several domains and to a specific target population. In any case the author should only need to add a little glue to link together the blocks found in the database. In other words a database should be as rich and as complete as possible with respect to a given domain.

8. MULTILINGUALITY

This factor is typical of the field of education and has no equivalent in the field of software. In no country is language a serious obstacle to software reusability, while educational material is usually dependent on a specific language and often on a specific national culture.
Differences in languages and cultures should not be a major obstacle to reusing the same design effort. This is especially important for European countries, which in order to obtain a market which is large enough, must

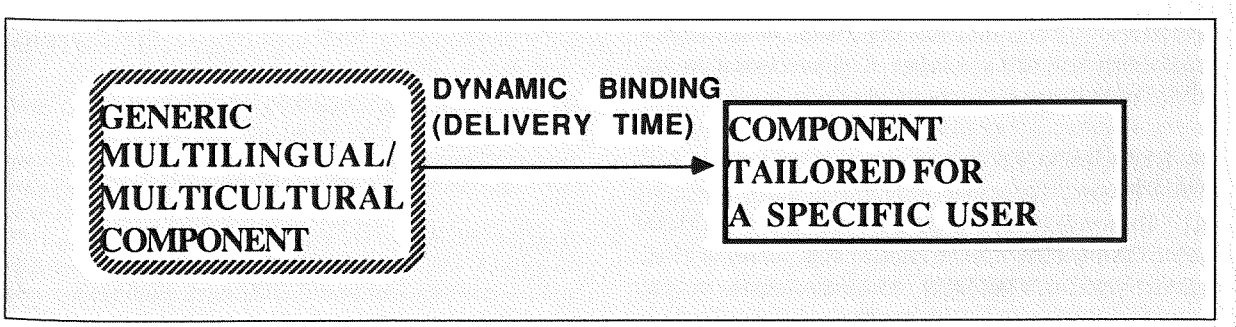


Figure 3: Tailoring a multilingual component for a specific user

think in terms of a European rather than national market.
A reusable component, when delivered to its final user, should refer to a specific language and to a specific culture. However, a reusable component, when designed, should be considered as a broad structure which can host equivalent multilingual or multicultural elements.
This means that, at the time of delivery, a generic multilingual component should be tailored to the specific linguistic background of the user by a sort of dynamic binding process (Fig. 3). This solution has interesting implications on the design process, which could be carried out by the author in a specific language and be extended almost automatically to all the languages represented in the database. Of course provision has to be made in the design process for different time/space requirements of equivalent language dependent elements.

9. CONCLUSIONS

From the above discussion it should be clear that the concept of reusability for educational software has several complementary components which make it very complex to put into practice. From the conceptual standpoint, most of these components are not yet fully clear and require further research both on the technical and on the pedagogical side. However we already have enough both conceptual and technological instruments, to start applying the concept of reusability to real world cases. Concrete experiences will be the best tool for refining existing concepts and methodologies.
The merging of multimedia and computers is a recent phenomenon; in the next few years it will be a widespread reality integrated into all pc's and workstations. Reusability will be a key factor to the success of the potential of multimedia in the field of education.

10. REFERENCES

[1] Burton R.R, Brown J.S., AN INVESTIGATION OF COMPUTER COACHING FOR INFORMAL LEARNING ACTIVITIES., Sleeman D. e Brow, J.S. (eds) Intelligence Tutoring Systems, Academic Press (London) Ltd, 1982

[2] Meyer B., OBJECT ORIENTED SOFTWARE CONSTRUCTION, Prentice Hall, 1988

[3] Olimpo G. et al., CONCEPTUAL MODEL FOR A DATABASE OF MULTIMEDIA LEARNING MATERIAL, Programme DELTA, Project D1012, Document D2, 1989

[4] Olimpo G. et al., MULTIMEDIA FLEXIBLE SYSTEMS AND ARTIFICIAL INTELLIGENCE, Proceedings of the Symposium "Advanced Research on Computers in Education ARCE", Tokio, Japan, 1990

[5] Olimpo G. et al., ON THE CONCEPT OF DATABASE OF MULTIMEDIA LEARNING MATERIAL, Proceeding of WCCE 90, Sidney, 1990

[6] Weyer S. A., AS WE MAY LEARN, Interactive Multimedia, Redmond, Washington, Microsoft Press, 1988

[7] Yankelovitch N., Smith K.E., Garrett L.N., Meyrowitz N., ISSUES IN DESIGNING A HYPERMEDIA DOCUMENT SYSTEM, Interactive Multimedia, Redmond, Washington, Microsoft Press, 1988

SUPER-LIBRARY: AN ENVIRONMENT FOR THE CONVERSION OF TEXT INTO HYPERTEXT

Flavio Argentesi, Nadia Catenazzi
KBS Laboratory
Joint Research Centre (JRC)
Ispra (VA), Italy

RIASSUNTO

Super-library è un ambiente progettato per supportare la conversione di documenti esistenti su carta in ipertesto. Il processo coinvolge due attività principali: l'identificazione delle componenti logiche nel testo sorgente e l'identificazione delle relazioni tra tali componenti. Il livello di automazione è fortemente dipendente dall'organizzazione dei documenti e dal dominio applicativo; il processo di conversione non può essere completamente automatizzato a causa delle implicazioni semantiche coinvolte; tuttavia si possono sviluppare strumenti per facilitare il suddetto processo.
Alcune caratteristiche rendono Super-Library qualcosa di più di un semplice ipertesto: l'interfaccia utente, il modello di navigazione, l'indicizzazione su tutto il testo e la possibilità di definire dei sinonimi forniscono un potente insieme di funzioni per localizzare ed accedere alle informazioni e facilitare l'orientamento dell'utente all'interno del documento.

ABSTRACT

Super-Library is an extensive environment designed to support the conversion of existing paper documents into hypertext. The process involves two main activities: the identification of the logical components in the source text and the identification of the interrelationships among these components.
The degree of automatic transformation is strongly affected by the organization of the documents and by the application domain; the conversion process can not be completely automated because of the semantic implications involved; however, tools can be developed to facilitate the process.
Some features make Super-Library an extended hypertext system: the user interface, the model of navigation, the full-text indexing and synonyms definition provide powerful functions for locating and accessing information and facilitate the user orientation in the document.

1. INTRODUCTION

Super-Library has two main goals: to facilitate the conversion of existing paper documents into hypertext, to create a flexible and rich environment in order to provide efficient retrieval facilities and an interface which can guarantee user orientation.
In section 2 we will describe some basic features of the system, i.e. possibility to manage a library of documents, multimedia capabilities, user interface and a model for document navigation.
In section 3 we will approach the problem of the conversion of text into hypertext; general steps of the conversion process are defined; specific capabilities of Super-Library are presented which enrich the basic hypertext structure: full-text indexing, synonyms definition and management.
In section 4 we will introduce and discuss an application which serves to test and improve the Super-Library techniques and to present important aspects of the system.

Section 5 will conclude the paper and will present some interesting ideas in the field of automatic generation of transversal links. The process will be approached as a knowledge engineering problem.

2. SUPER-LIBRARY: AN EXTENDED HYPERTEXT ENVIRONMENT

As starting point Super-Library makes reference to the Superbook project [2].
It deviates from Superbook on the following points:
1) it is designed to manage a library (a large group of interrelated documents, indexes and semantic references are treated as a whole);
2) it considers the problem of defining transversal links;
3) it also provides full multimedia support (graphics, pictures, text);
4) the user interface is developed starting from Superbook interface ([2], [8]) and provides all the extensions required to cope with multiple documents and multimedia management.

The Super-Library interface consists essentially of three windows (see *figure 1*):

- the Text Window, which contains the currently selected page of text; embedded links are put in evidence by special active markers, search terms are automatically highlighted; it can hold high resolution pictures and graphics;
- the Search Window to define Boolean search queries;
- the Dynamic Table of Contents Window; documents are hierarchically organized; users can dynamically navigate in the table; they can open an entry up to the desired depth, maintaining higher level information visible for contextual orientation. When the user selects a Table of Contents entry, representing an item at the lowest level of the hierarchy, the corresponding text is automatically displayed in the Text Window.

The Table of Contents Window is also used to show the results of a search: the frequency of occurrences of search terms is posted next to each entry (see *figure 2*). At this moment user can select a text in a similar way as described above. This is a more efficient way for the user to access the terms searched for, because the results of a search are shown in the context of the Table of Contents and not in a list of page numbers [2] or as the next occurrence of the search term from the current position in the text.
Other windows specify: the topic heading of the displayed page and the title of the current document; a menu with a complete list of the available documents allows to open another document and to show the corresponding Table of Contents in the appropriate window.
Typical text navigation functions such as "next page", "previous page", "back at the last page", complete the configuration.

This type of interface provides users with the possibility to navigate, maintaining the sense of location and direction with a low cognitive overhead, partially solving a common problem of the hypertext systems: the disorientation problem ([1], [7]).

3. THE CONVERSION OF PAPER DOCUMENTS INTO HYPERTEXT

The development of conversion techniques for existing documents into hypertext is a new interesting research area; not all applications are appropriate for this transformation. Specific characteristics are required: the original body of information has to be organized into fragments, the fragments are related to each other, the user needs only a small fraction of them at any time [5].
The problem of converting text into hypertext is a very difficult task to be automated, the reason being the need of an extensive human assistance because of the semantic implications involved.
A definition of the steps of the conversion process is given in Furuta, Plaisant and Shneiderman [5] .
The first problem to be addressed is the identification of the logical components of the source text; normally it consists of capturing the hierarchical structure (nodes correspond to paragraphs in the text) which is text intrinsic and is typically represented by markers (SGML). A parsing program is developed to extract the components and to create the basic structure of the hypertext [4]: nodes and hierarchical links.

The second problem is the definition of transversal links. The automation of the design of the links network requires special care because incorrectly-formed links as well as not created links destroy the integrity of the hypertext. The automation process can only suggest a

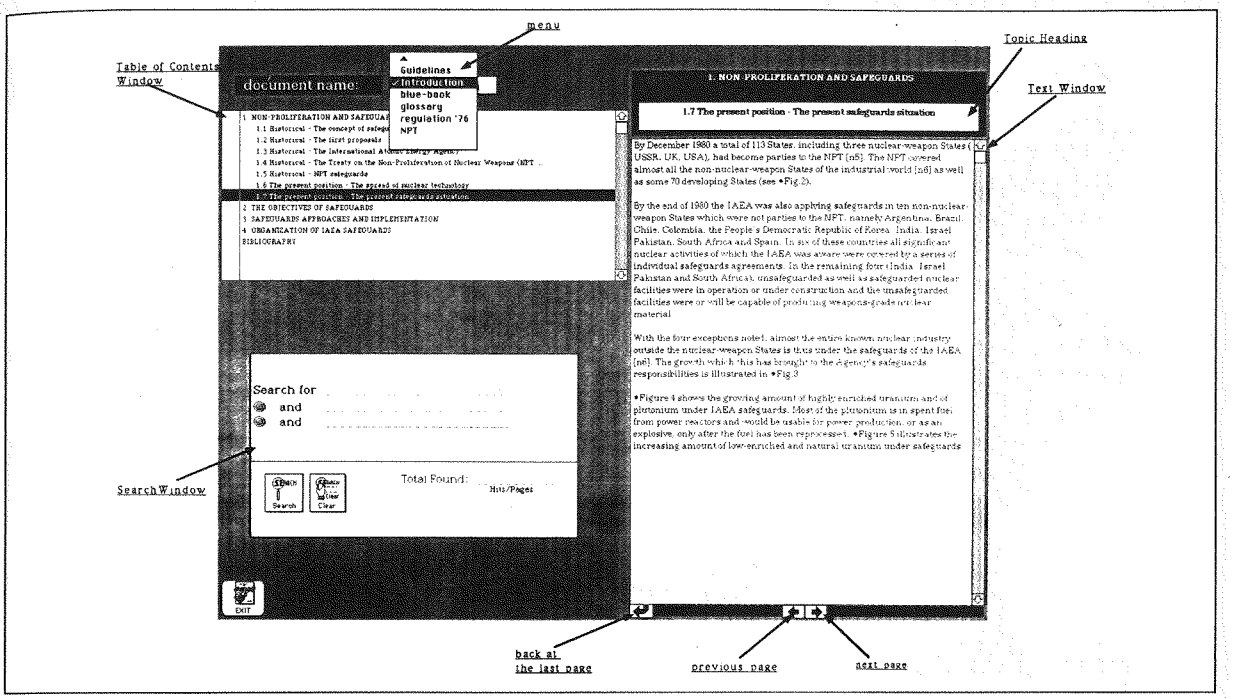


Figure 1

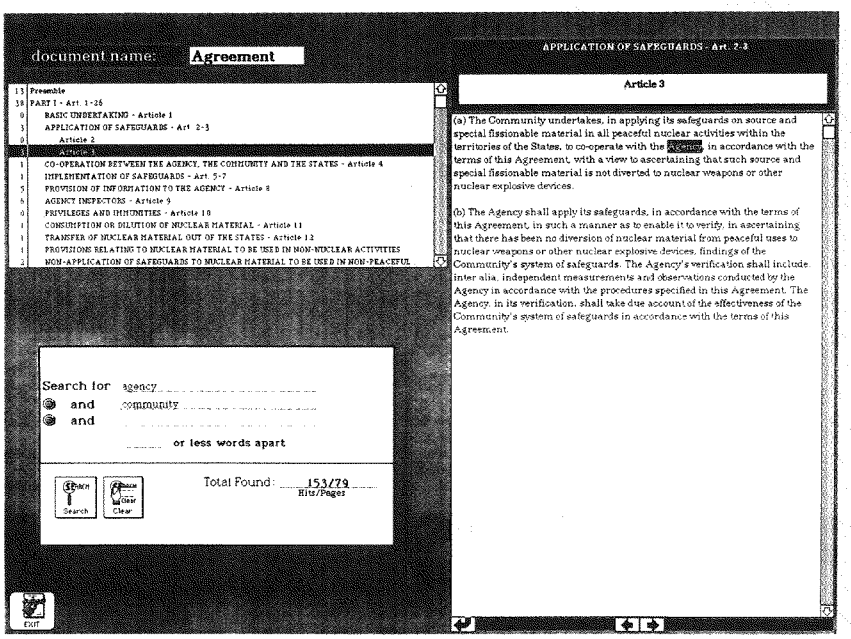


Figure 2

certain number of transversal links which have to be accepted by the user. This is why we need human intervention.
Other addressed problems are useful to complete the system: the addition of new nodes to provide indexes, the reformatting of the text with indentations, different fonts, etc., the addition of relevant material such as preface, introduction, summary, etc.

These are the basic steps of the conversion process. Super-Library expands this model providing a series of new capabilities: one of these is the full-text indexing (every occurrence of every word in the document, except stop-words, is indexed).
Automatic tools for indexing have been integrated within the chosen hypertext platform (Hypercard): indexes can be built for the single document and for the library permitting a simultaneous search on the whole documentation; the system offers a range of search criteria available individually or in combination and may use the Boolean AND, OR, and BUT NOT connectives: individually word search, adjacent word search, word search using truncation, exact order search make the search process very powerful.
We intend to add the possibility for users to define their search synonyms, i.e. terms that will be associated to words used by the author of the document [2], and constitute in this way a thesaurus.
Every word acts as a "hot link" to other potential related parts of the text; users can browse through a document without following the organization schema (full-text indexing) and terminology (synonyms) proposed by the author in order to get the required information; this is a big limitation of the conventional paper documents and of many hypertext systems: the lack of a thesaurus leads to a very poor search performance (poor recall) although the target information is available.

4. THE TEST PROBLEM: DOCUMENTATION ON THE IAEA SAFEGUARDS SYSTEM

A group of interrelated hierarchical documents in the area of Nuclear Safeguards is used as test application for the system.
This work is part of a more general project "ISS" (Intelligent Safeguards System) that deals with the control and the management of nuclear material. It concerns the design and implementation of a system able to manage all the related documentation: Treaties, Regulations and Informative Booklets. Appendix 1 contains a description of the contents of each document.
We will first outline the basic objectives and role of the Safeguards System, then we will consider this application as a test problem for the system described above.

4.1 The IAEA Safeguards System

Safeguards are essentially the technical means of verifying the fulfilment of political obligations undertaken by different States in concluding international agreements relating to the peaceful use of nuclear energy.
Today most of these obligations flow from the NPT (Treaty on the Non-Proliferation of Nuclear Weapons) and similar agreements.
The International Atomic Energy Agency (IAEA) in Vienna is an organization responsible for the application of the non-proliferation policy; it verifies that no significant quantity of nuclear material is diverted from peaceful activities.
The IAEA safeguards approach is based on nuclear material accountancy as fundamental safeguards measure; the purpose is to establish the quantities of nuclear material present within defined areas and the changes in these quantities; these activities are undertaken by the operators of the facilities in the State and by a Safeguards Authority (IAEA inspectors) who verify the information received by the State.
In Europe safeguards operations, which are under the control of IAEA, are delegated to the DCS (the Control of the Security of Euratom) in Luxemburg.
The ISS is a complex system aimed at the support of all DCS operations; it will be used for purposes such as storage, retrieval and checking of accountancy and inspection data, comparison of accountancy and inspection data, report generation and automatic distribution, surveillance support (collecting video data), storage and access to related documents.

4.2 Conversion process for IAEA documents

The documents were already available in a computer readable form; few manipulations were applied to the original text. The criterion used to define the logical components is suggested by the documents structure: a component could be:
- an article, for books such as the Treaty, the Agree-

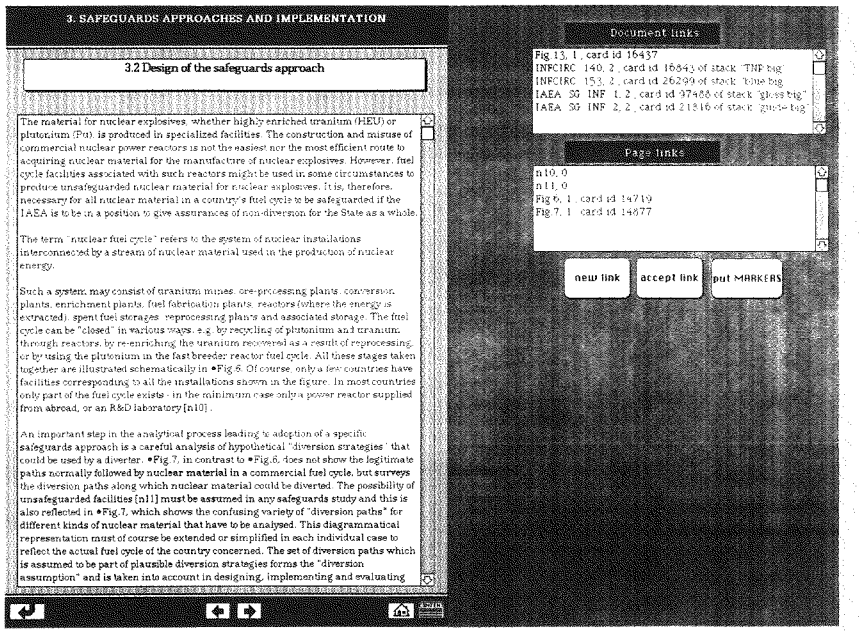


Figure 3

ments,
- a term, for the glossary,
- the smallest unit having an heading, for the other documents.

A special purpose program has been implemented which takes as input the original text in ASCII and builds the basic hypertext structure.
Figures were acquired separately with a scanner and integrated in a second time; a particular treatment was made to include footnotes and figures captions and to manage links inside figures or tables; the text was slightly reformatted to improve the readability: better identation, empty lines, etc.

Specific tools are developed for definition of transversal links; I identify different types of links arising from explicit references in the text to:
- bit map images (figures or tables);
- other part of text (in the same or in another document);
- notes;

These tools allow the author to define: the starting point of the link (simply selecting a string in the text), the type of the link, the level (we will explain later), the end point of the link; the button "new link" in *figure 3* activates this process; a confirmation has to be done with the button "accept link": the system controls that the same starting point has not previously been used and saves the new link properties in a list which contains all the links information; the button "put markers" inserts markers in the text to show the active words, i.e. the link starting points.
Links are associated to words or strings, not to the terms position in the page; this reduces redundancy when multiple occurrences of the same term exist; but it could produce some problems when the same word has a different meaning in a different context; the adopted solution is to use two levels: the document level and the page level; terms having always the same meaning are put at the document level; the page level is used for terms valid only for one page or having a different end point in comparison with the document level.
The indexing process is automatic; it only requires the definition of text(s) on which it has to operate and, in a second time, to associate the index to the document name so that transparency for users is achieved during the browsing phase; changing the document causes an

automatic change of the index.

When the preprocessing is complete, documents are accessible through the configuration display described above. Figure 1 shows the typical display: a partially expanded Table of Contents, a page of text loaded in the window with markers for "active words". Figure 2 represents the effects of a search: the results are directly visible on the Table of Contents Window and on the Text Window.

5. CONCLUSION

It clearly results from this experience that a fully automatic conversion of paper-document into hypertext is difficult and not always achievable; the process is easier when "documents are regularly and repetitively structured" [5], but requires a deep study of documents and the domain area. We do not agree with Egan's opinion, when he states that "the purported advantages of human intervention in developing hypertext systems have not been demonstrated". For our application the human assistance was indispensable, although automatic tools proved to be useful for some tedious activities such as loading the source document, creating the indexes.
While the construction of hierarchical hypertext structure is quite easy to automate, special care is required for the definition of transversal links; in our future studies we intend to focus our interest in this direction: our idea is to build the hypertext network incrementally by successive definitions of semantic layers; the scope is the generation of a backbone of transversal links for further editing and extensions by users. The link generation will be proposed as a knowledge-based problem solving task.
The process will consist of: the identification of concepts relevant to the application area, the association of them to the hypertext nodes, describing nodes in terms of these concepts, the definition of a set of rules for connecting nodes in different semantic layers. Domain experts will be required to build the metalevel and define rules and knowledge structure.

6. REFERENCES

[1] J. Conklin, HYPERTEXT: AN INTRODUCTION AND SURVEY, Microelectronics and Computer Technology Corp., IEEE Computer 20, Sept. 1987

[2] D.E.Egan, J.R. Remde, L.M. Gomez, T.K. Landauer, J. Eberhardt, C.C. Lochbaum, FORMATIVE DESIGN-EVALUATION OF SUPERBOOK, ACM Transactions on Information System, Vol. 7, No.1, January 1989

[3] M. Frisse, M. D., SEARCHING FOR INFORMATION IN A HYPERTEXT MEDICAL HANDBOOK, Washington University School of Medicine, Communications of the ACM, July 1988

[4] M. Frisse, S. B. Cousins, INFORMATION RETRIEVAL FROM HYPERTEXT: UPDATE ON DYNAMIC MEDICAL HANDBOOK PROJECT, St. Louis, Missouri, Hypertext '89 Proceedings, November '89

[5] R. Furuta, C. Plaisant, B. Shneiderman, A SPECTRUM OF AUTOMATIC HYPERTEXT CONSTRUCTIONS, Department of Computer Science, University of Maryland, HYPERMEDIA, Vol. 1, Num. 2, 1989

[6] D. R. Raymond and F. Wm. Tompa, HYPERTEXT AND THE OXFORD ENGLISH DICTIONARY, University of Waterloo, Hypertext '87 Proceedings, November '87

[7] K. Utting, Y. Yankelovich, CONTEXT AND ORIENTATION IN HYPERMEDIA, Brown University, ACM Transactions on Information System, Vol. 7, No.1, January 1989

[8] J. H. Walker, DOCUMENT EXAMINER: DELIVERY INTERFACE FOR HYPERTEXT DOCUMENTS, Symbolics Inc., Cambrigde Center, Cambridge, Hypertext '87 Proceedings, November '87

References to the IAEA documents:

[9] Commission of the European Communities, THE AGREEMENT BETWEEN EURATOM AND THE IAEA IN IMPLEMENTATION OF THE TREATY ON THE NON-PROLIFERATION OF NUCLEAR WEAPONS, 1973

[10] International Atomic Energy Agency, IAEA SAFEGUARDS - Glossary (Rev.1), Vienna, 1987

[11] International Atomic Energy Agency, IAEA SAFEGUARDS - Guidelines for States' Systems of Accounting for and Control of Nuclear Materials, Vienna, 1980

[12] International Atomic Energy Agency, IAEA SAFEGUARDS - An Introduction, Vienna, 1981

[I3] International Atomic Energy Agency, THE STRUCTURE AND THE CONTENT OF AGREEMENTS BETWEEN THE AGENCY AND STATES REQUIRED IN CONNECTION WITH THE TREATY ON THE NON-PROLIFERATION OF NUCLEAR WEAPONS, June 1972

[14] International Atomic Energy Agency, SAFEGUARDS

[15] Commission of the European Communities, COMMISSION REGULATION (EURATOM) OF 19 OCTOBER 1976, in Official Journal of the European Communities, Vol. 19, 31-12-76

Appendix 1

The documents used for the application are: three Informative Booklets (an introduction, a glossary and guidelines for the control of nuclear material), the NPT, the "BLUE-BOOK", the Agreement between EURATOM and IAEA, the Commission Regulation.

The Introduction [IAEA/SG/INF/3] provides a picture of the objectives and methods of IAEA safeguards and their role in the non-proliferation policy.

The Glossary [IAEA/SG/INF/1] is aimed to facilitate the understanding and standardization of the meaning of the safeguards terminology within the international community. The terms have not been arranged in alphabetic order but they are collected following the internal relation of the subjects treated; the organization of the document results hierarchical.
The "Guidelines for States' Systems of Accounting for and Control of Nuclear Materials" (SSAC) [IAEA/SG/INF/2] presents guidelines for the organization and functions of SSAC with respect to obligations arising from Safeguards Agreements concluded by a State with the IAEA.

The United Nations' NPT [NPT] is the most important political instrument for preventing the spread of nuclear weapons and it is the reference document for the IAEA safeguards system. It entered into force in 1970. It states:
"Each nuclear weapon State party to the Treaty undertakes not to transfer nuclear weapons or other explosive devices, and not to support manufacture or acquisition of such weapons by any non-nuclear weapons States";
"Each non-nuclear weapons State undertakes not to acquire nuclear weapons or other explosive devices; they also bind themselves to conclude an agreement with the IAEA for the application of safeguards";
in return, the Treaty recognizes the right of all Parties to participate in the full exchange of equipment, material, information on peaceful uses of nuclear energy. The Parties also undertake to pursue negotiations towards nuclear disarmament.
The "BLUE-BOOK" is an IAEA document entitled: "The structure and the content of agreements between the Agency and States required in connection with the Treaty on the Non-Proliferation of Nuclear Weapons" [INFCIRC/153]; the agreement contains, according with article 3 of the Treaty, an undertaking by the State to accept safeguards on all source or special fissionable material in all peaceful nuclear activities for the purpose of verifying that such material is not diverted to nuclear weapons.
The BLUE-BOOK is the standard NPT safeguards agreement; substantial additions to the standard text have occurred in the case of a group of States (Euratom States), which are performing specific safeguards activities (DCS) in their territories.
The specific document, included in the list, is the Agreement between Euratom and the IAEA in implementation of the NPT (1973) [9].

The Commission Regulation (EURATOM) of 19 October 1976 [15] concerns the application of the provisions on Euratom Safeguards; it's a practical description on the modality of applying safeguards in the Euratom States. It is published on the Official Journal of the European Communities.

HYPERCHIP: A SYSTEM ENGINEERING ENVIRONMENT WITH HYPERTEXTUAL FRONT-END

Stefano Antoniazzi, Mirella Mastretti
Central Research Laboratories, ITALTEL SIT
Castelletto di Settimo Milanese (MI) - Italy

RIASSUNTO

Hyperchip è un prototipo di ambiente interattivo il cui obiettivo è proporre un nuovo approccio metodologico alla progettazione di sistemi hardware e software. Particolare enfasi è stata attribuita alla modellazione ad alto livello e all'applicazione di tecnologie software innovative quali interfacce utente ipertestuali e programmazione object-oriented in ambito CAD. Il processo di progettazione può essere considerato, infatti, come navigazione in un ipertesto di rappresentazioni così come un sistema ad ipertesti supporta la navigazione in un data-base testuale o grafico. Il concetto di ipertesto può suggerire la base di un nuovo paradigma di interazione tra utente e data-base in ambienti di system-engineering. In particolare, gli ipertesti introducono il concetto fondamentale di "astrazione" e le possibli relazioni fra oggetti astratti e grafica.

ABSTRACT

HYPERCHIP is a research prototype intended to propose a new methodological approach to the design of hardware and software systems with emphasis on high-level modeling and to explore innovative software technologies, such as hypertext user interfaces and object-oriented programming, in the framework of computer-aided design. The design process can be considered as a navigation in a hyperspace of representations, in the same way as a hypertext system supports the navigation in a textual or graphical database. The concept of hypertext can suggest the basis of a new paradigm for user-database interaction in system engineering environments. In particular, hypertext introduces the fundamental concept of "abstraction" and possible relationships between abstract objects and graphics.

1. INTRODUCTION

A new generation of tools is needed to support the increasing complexity of the design process, in particular for hardware and software systems.
The ultimate goal, in our opinion, should be based on the following key requirements:

- full integration of tools through a unified environment (design framework)

- better integration between user and design tools

- more suitable data models (object-oriented databases)

- full automation (wherever possible) of each process, translating the human intervention to more abstract and creative tasks

- availability of tools for correctness verification, performance evaluation and "what-if" analysis at each level of abstraction.

HYPERCHIP is a research prototype intended to propose a methodological approach to the design process with emphasis on high-level system modeling, and to explore innovative software technologies, such as object-orien-

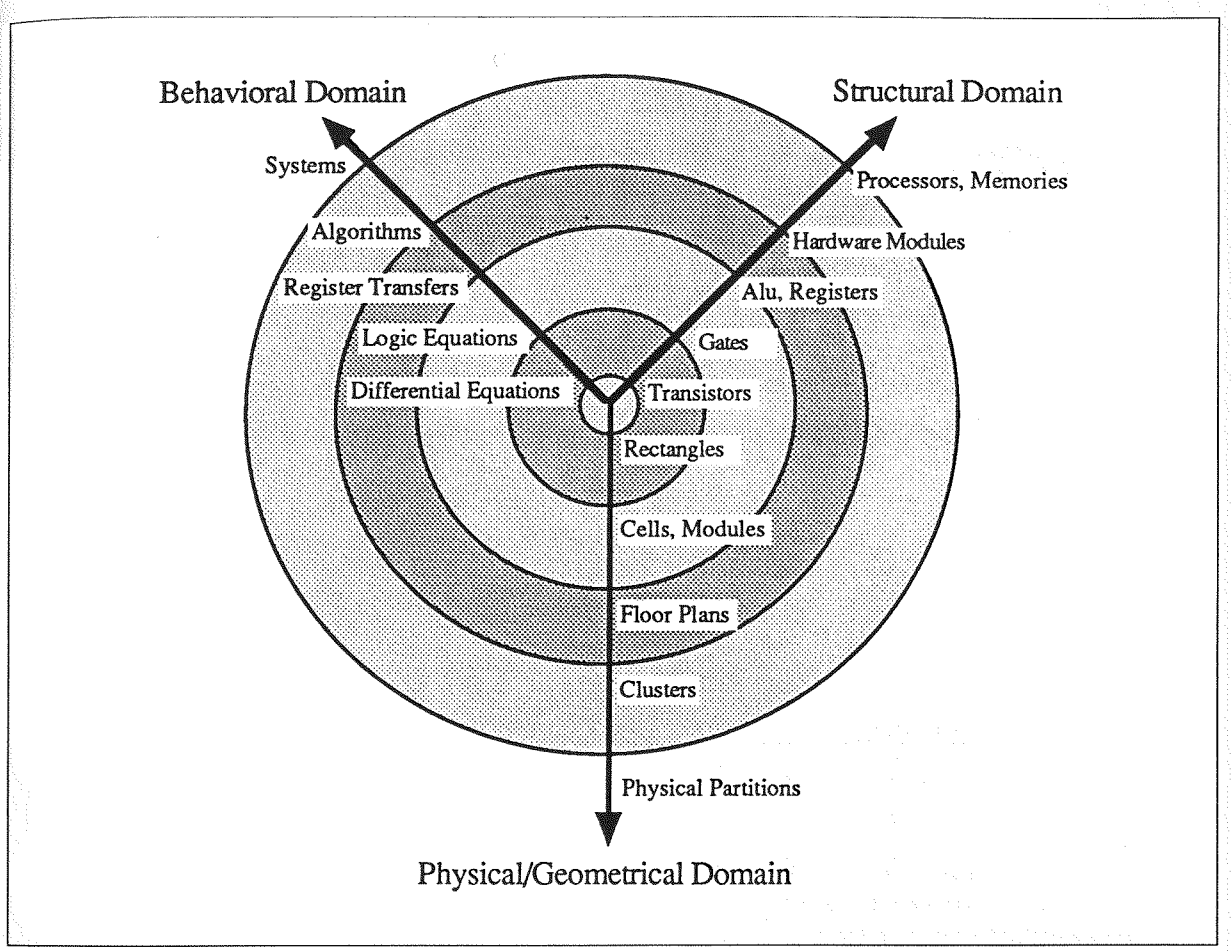


Figure 1: Design Hyperspace

ted programming and hypertext user interfaces, in the computer-aided design area.
Although the first version of HYPERCHIP is mainly concerned with VLSI design tasks, the application range of the proposed methodology can be generalized to support mixed hardware/software systems.
The first version of the prototype was built in the Smalltalk environment.
Currently, for our research work we have chosen a more industry-oriented development platform based on UNIX workstations networks, the C++ language and the standard X-Window/Motif graphical environment.

2. NAVIGATION IN THE DESIGN HYPERSPACE

The design process for hardware and/or software systems is a very complex activity.
To approach the problem from a theoretical point of view, D.Gajski has proposed a graphical representation named "design hyperspace" (Fig.1).
This chart shows three orthogonal domains.
The behavioral domain is only concerned with the functional specification of a system.
This domain captures the "essential model" of a system with no reference to implementation details.
In the structural domain a system is described as a network of connected entities.
It forces an architectural partitioning but the description is still uncommitted to a specific technology.
Technological details, such as layout for VLSI chips or PC boards, are handled in the physical domain.
The relationship between behavior and structure, as well as between structure and physical representation,

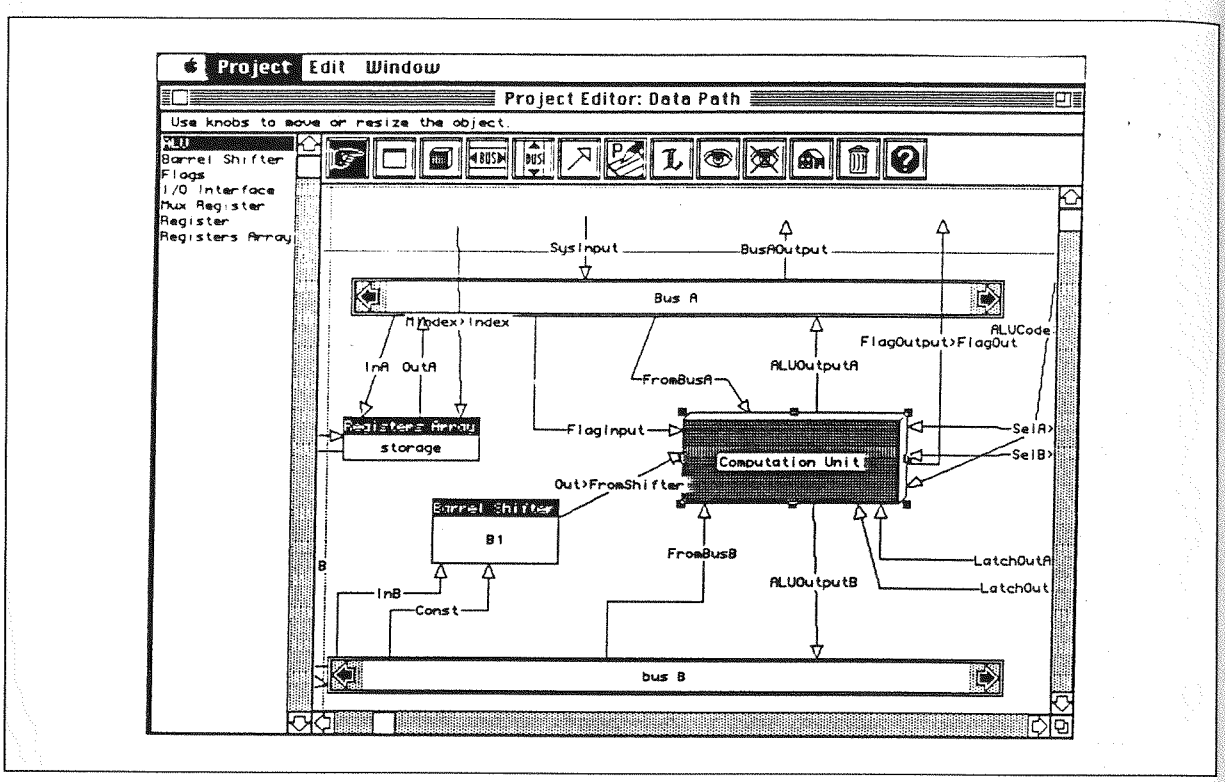


Figure 2: Visual Browser

is in general a one-to-many mapping where a lot of known or unknown alternative paths are possible, so trade-off problems are very common.
Moreover, each domain can be explored at many different levels of abstraction.
Therefore the design process can be considered as a navigation in a hyperspace of representations, in the same way as a hypertext system supports the navigation in a textual or graphical database.
Each design representation can be associated with a graphical view.
Conventional VLSI/CAD systems already apply this idea to the lower levels of design (schematic entry, geometrical layout for full custom ICs).

Our goal is to extend this approach to the architectural level and the functional specifications.
The concept of hypertext can suggest a new paradigm for user-database interaction in system engineering environments.
In particular, hypertext introduces the fundamental concept of "abstraction" and possible relationships between abstract objects and graphics.

3. A VISUAL BROWSER FOR SYSTEM SPECIFICATION

In HYPERCHIP, at the upper level of abstraction, the user has to enter a model for the system to be designed. The model is built through a visual editor/browser (Fig.2).

This tool supports an object-oriented and direct-manipulation graphical interface.
We have also designed an original auto-routing algorithm, so a user should specify only two related objects, while the graphical connection is fully automated.
The same algorithm automatically handles connectivity information when objects are moved or resized.
After the selection of an object with the mouse, the user can request a large set of actions. He can change parameters, open a different view of the object (for example a structural or physical view), associate a behavior, and so on.

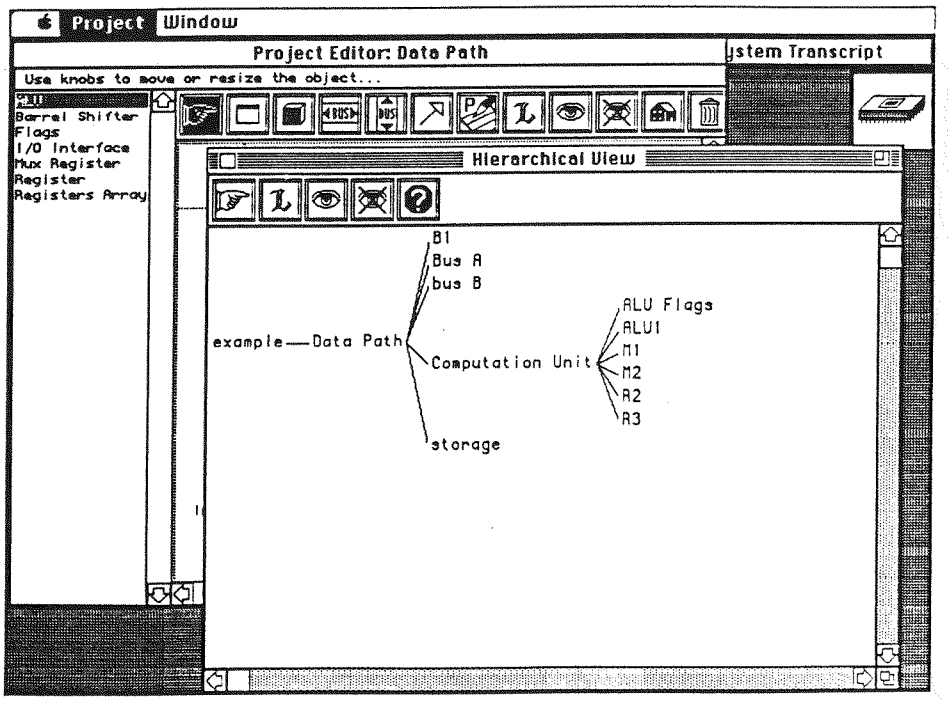


Figure 3: Hierarchical View

All the views of an object are synchronized through the Model-View-Controller (MVC) paradigm.
The views share a common database, in this way a change in a view automatically updates related views through a specific message-passing protocol.
The designer interactively describes the system as a hypergraph whose nodes can be either primitives or "black boxes". Edges model the message flow between nodes.
A special view (Fig.3) can be opened to show graphically the hierarchical structure of the model.
This kind of description has also an object-oriented nature.
Each node is an instance of a parameterized class and all classes are stored in specific libraries.
An instance keeps its private information, such as values for parameters and references to related instances.
Each primitive node is associated with a timed finite-state-machine behavioral model, and it always represents a leaf in the hypergraph hierarchy.
The behavior is shared by all instances of the same class. Instances communicate through message-passing.
A black box node is intended to be expanded with a lower level subgraph and so on, recursively, until leaves are reached.
Black boxes are so general that any kind of system can be described in an a very abstract way.
The visible directed edges propagate messages between nodes at the same hierarchical level. Invisible links handle the bi-directional message flow between a black box and its subgraph nodes.
While point-to-point communication has a more natural graphical representation, broadcasting is also allowed through special primitives, called "buses" or "mailboxes".

An important concept here is that edges have no physical semantics, they are not "wires" as in conventional CAD schematic entry tools.
While this kind of representation can be used for architectural or register-transfer level modeling, it is more attractive for very abstract descriptions at the functional specification level.

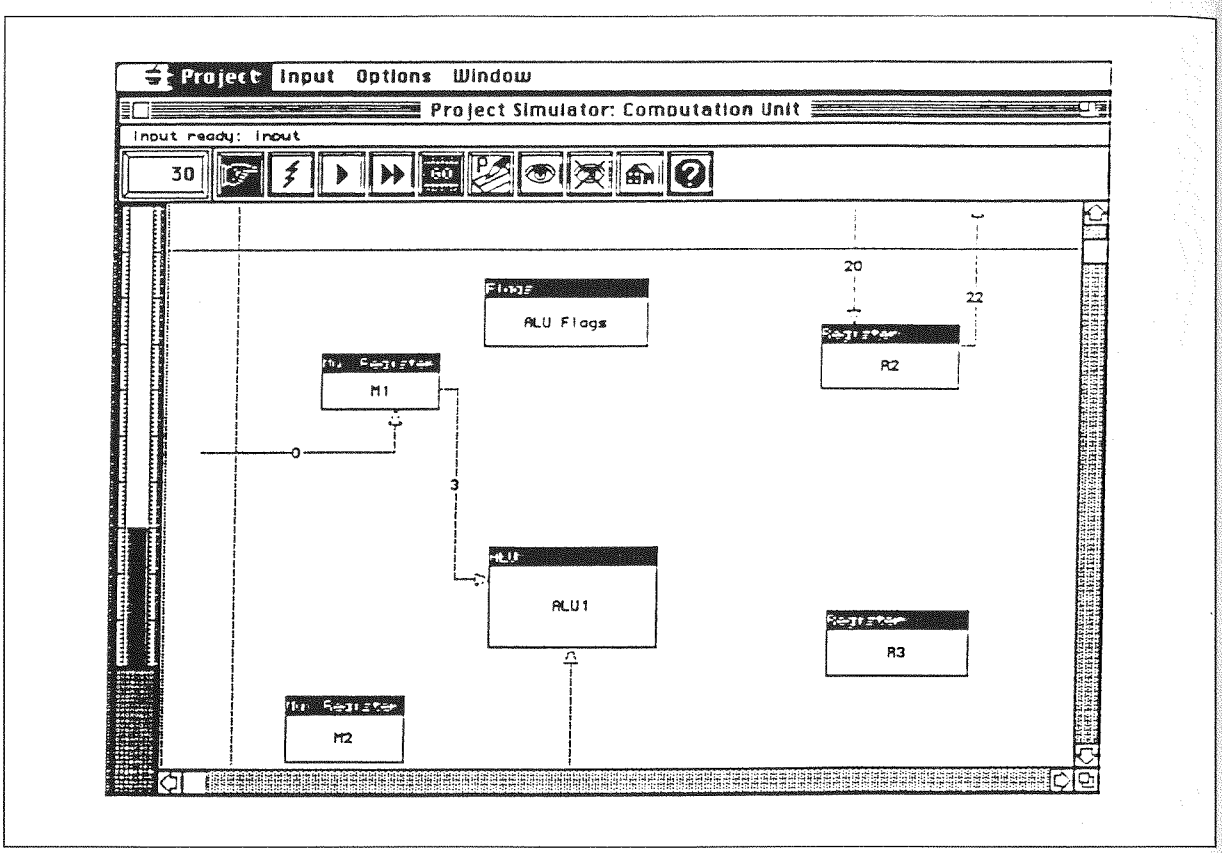


Figure 4: Visual Simulator

Therefore we should not attribute a structural semantics to this description in any sense.

4. EXECUTABLE SPECIFICATIONS AND GRAPHICAL ANIMATION

A data-driven semantics was chosen for the hypergraph model built with the visual editor.
A data flow graph is similar to a Petri network. A node can fire when all the input edges have tokens (data items) and no tokens are placed on output edges.
After firing, tokens are removed from all input edges and are placed on all outputs.
The advantage of this formalism is that the visual hypergraph representation also allows an execution (simulation).
HYPERCHIP supports both symbolic and functional simulation processes.
During symbolic simulation tokens have no specific associated data.
The simulator plays the "token game" using the hypergraph description.
This process is useful to collect preliminary statistics of performance without putting a detailed behavioral description into each node.
Important parameters of performance are, for example, utilization and latency.
Through symbolic simulation the user can quickly localize possible bottlenecks in the system model.
Functional simulation is more precise but it requires a behavioral description for each entity.
This kind of analysis is required when data-dependent behaviors must be modelled.
In its first version, HYPERCHIP supports behavioral descriptions written as Smalltalk code and finite-state-machine tables.
A particular feature of the HYPERCHIP simulator is that the hypergraph is not flattened during the execu-

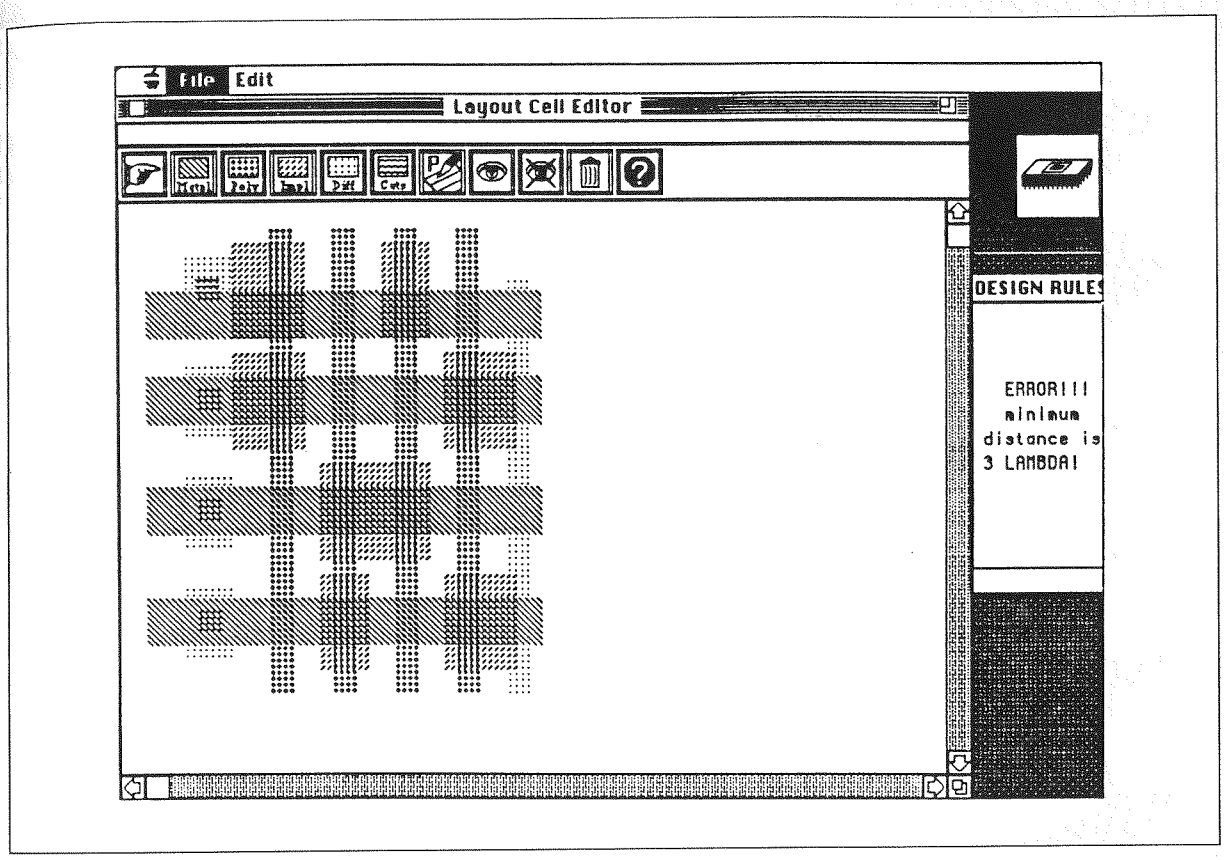


Figure 5: Layout Editor

tion, but it keeps its hierarchical structure.
Another very important point is the timing model.
In hardware design, as well as in real-time software design, a precise timing control is essential.

For this reason, the simulator uses time delay parameters, local to each node, for a correct scheduling of firing events.
In HYPERCHIP the simulation process is visual and interactive (Fig.4).
The user can control the execution through an iconic menu supporting useful options such as single-step, programmable step, continuous mode and so on.
Each simulation event has a visual feedback.
Following a hypertext approach, objects such as nodes or edges can be selected with the mouse to get simulation statistics or history information.
The execution is carried out simultaneously at each level of the hierarchy.
Through an explode/implode mechanism the user can open or close black boxes.
In this way one can navigate in the model and explore the behavior of any particular subsystem.

5. LINKS TO LOGIC SYNTHESIS AND PHYSICAL LAYOUT

HYPERCHIP was built as an open system and integrated environment.
For this reason a link is provided to other lower level design tools, to support, for instance, logic synthesis and physical layout processes.
To demonstrate a possible integration, we have included in HYPERCHIP a layout editor for Manhattan-style full custom chips in NMOS technology (Fig.5).
Geometrical data are automatically derived through cell generators and are manually modifiable.
Design rules check is supported though message-pas-

sing and constraint propagation techniques.
The main purpose of this tool is to show a general object-oriented approach to all levels of abstraction in system design.

6. ACKNOWLEDGEMENTS

We wish to thank HUG (Hypertext User Group) and, in particular, Prof. Gianni Degli Antoni (Department of Computer Science - Milan University) for his useful suggestions.

7. REFERENCES

[1] Jay Banerjee, W.Kim, H.Chou, J.Garza, D.Woelk, COMPOSITE OBJECT SUPPORT IN AN OBJECT-ORIENTED DATABASE SYSTEM, OOPSLA '87

[2] Jay Banerjee, W.Kim, H.Chou, J.Garza, D.Woelk, N.Ballou, DATA MODEL ISSUES FOR OBJECT-ORIENTED APPLICATIONS, ACM Transactions on Office Information Systems, January 1987

[3] Edward Barrett, TEXT, CONTTEXT AND HYPERTEXT , MIT Press, 1988

[4] James Bigelow, HYPERTEXT AND CASE, IEEE Software, March 1988

[5] Raymond T.Boute, SYSTEMS SEMANTICS: PRINCIPLES, APPLICATIONS AND IMPLEMENTATION, ACM Transactions on Programming Languages and Systems, January 1988

[6] R.Buhr, G.Karam, C.Hayes, C.Woodside, SOFTWARE CAD: A REVOLUTIONARY APPROACH, ACM Transactions on Software Engineering, March 1989

[7] Shi-Kuo Chang, VISUAL LANGUAGES: A TUTORIAL AND SURVEY, IEEE Software, January 1987

[8] Brad J. Cox, OBJECT-ORIENTED PROGRAMMING: AN EVOLUTIONARY APPROACH, Addison-Wesley, 1986

[9] E.F.Deprettere, J.Bu, ANALYSIS AND MODELING OF SEQUENTIAL ITERATIVE ALGORITHMS FOR PARALLEL AND PIPELINE IMPLEMENTATIONS, ISCAS '88

[10] Tom De Marco, STRUCTURED ANALYSIS AND SYSTEM SPECIFICATION, Yourdon Press, New York 1978-1979

[11] D.Gajski, SILICON COMPILATION

[12] Pankaj K. Garg, ABSTRACTION MECHANISMS IN HYPERTEXT, Communications of the ACM, July 1988

[13] Joseph A. Goguen, J.W.Thatcher, E.G.Wagner, AN INITIAL ALGEBRA APPROACH TO THE SPECIFICATION, CORRECTNESS AND IMPLEMENTATION OF ABSTRACT DATA TYPES, Current Trends in Programming Methodology, vol.4, Prentice-Hall, 1978

[14] Joseph A. Goguen, PARAMETERIZED PROGRAMMING, IEEE Transactions on Software Engineering, N.5, September 1984

[15] HYPERTEXT '87, Proceedings, University of North Carolina, March 1988

[16] K.M.Kavi, B.Buckles, N.Bhat, ISOMORPHISMS BETWEEN PETRI NETS AND DATAFLOW GRAPHS, IEEE Transactions on Software Engineering, October 1987

[17] B.Kruatrachue, Ted Lewis, GRAIN SIZE DETERMINATION FOR PARALLEL PROCESSING, IEEE Software, Januray 1988

[18] Barry D. Kurtz, Donna Ho, Teresa A.Wall, AN OBJECT-ORIENTED METHODOLOGY FOR SYSTEMS ANALYSIS AND SPECIFICATION, Hewlett-Packard Journal, April 1989

[19] Bertrand Meyer, OBJECT-ORIENTED SOFTWARE CONSTRUCTION, Prentice Hall, 1988

[20] Rami R.Razouk, THE USE OF PETRI NETS FOR MODELING PIPELINED PROCESSORS, 25th Design Automation Conference, 1988

[21] Arthur H.Veen, DATAFLOW MACHINE ARCHITECTURE, ACM Computing Surveys, December 1986

[22] Paul T.Ward, Stephen J.Mellor, STRUCTURED DEVELOPMENT FOR REAL-TIME SYSTEMS, Yourdon Press, New York, 1985

HYGRAPHS FOR HYPERTEXT MODELLING*

Carlo Alberto Bosello, Fabio Vitali
Computer Science Laboratory
University of Bologna

RIASSUNTO

In questo lavoro si tratteggia brevemente la struttura matematica sottostante alcuni sistemi ipertestuali, e si riassumono le principali caratteristiche di RHYTHM (Research on Hypertext Theory and Management), il progetto ipertestuale attualmente in sviluppo al Laboratorio di Scienze dell'Informazione dell'Università di Bologna. Seguendo il lavoro di Harel ([Har88]), viene introdotta una struttura derivata dal grafo, chiamata higrafo, e ne viene discussa la convenienza nel rappresentare vari oggetti ipertestuali. Viene discusso inoltre un arricchimento dell'higrafo, chiamato s-higrafo, e ne viene spiegato l'uso in RHYTHM. Viene infine data una specificazione degli higrafi ed s-higrafi come oggetti algebrici.

ABSTRACT

In this paper we briefly outline the underlying mathematical structure of several hypertext systems with a short summary of RHYTHM (Research on Hypertext Theory and Management), the hypertext project presently under development at the Computer Science Laboratory of the University of Bologna. A graph-like structure called higraph is introduced following the work of David Harel [7], and its suitability for the representation of various hypertextual objects is discussed. An enrichment of higraphs, s-higraphs, is also discus-sed along with its use in RHYTHM. A specification of higraphs and s-higraphs as algebraic objects is finally given.

* This work has been done within the subproject #7, "Sistemi di supporto al lavoro intellettuale" of the Progetto Finalizzato CNR "Sistemi informatici e Calcolo Parallelo".

1. INTRODUCTION

Hypertext systems (h.s. in the following), born to augment and free human capabilities, are nowadays suffering from many of the drawbacks they were created to solve. The rigid, graph-based structure of nodes, the insurmountable separation of passive users (readers) and active users (authors), the use of many different data formats are clear evidences of this fact.
RHYTHM was therefore born to outdo these problems, to be slightly more than a hypertext, to be rather a system for hypertext publishing, i.e. a system which is not to be limited to didactic or tutorial ambitions, but allowing support of intellectual work of all kinds.

We thought that graph theory was not enough for our aims, so we found an enrichment of it in Harel's work, *higraphs* [7]. They allow both binary relations (or n-ary, if we talk about hypergraphs) and set-theoretical relations, like union, intersection, etc., and, like graphs, have an immediate graphical visualisation.

Three are the implementations of higraphs to be studied in RHYTHM: groups, or arbitrary sets of users, *networks*, arbitrary sets of machines, and mainly *documents*, arbitrary sets of bytes.
This need to have higraphs right at the foundations of

RHYTHM, and not simply as an informal psychological model, and the willingness to precisely isolate and qualify the base objects that ought to constitute RHYTHM led us to propose a definition of higraphs as *abstract data types*, and to give, after Ehrig and Mahr [4] terminology, its *algebraic specification.*
It then became clear that - in at least one case - not even higraphs were enough: documents are not a shapeless agglomerate of bytes, but they own an internal linear structure that higraphs could not cope with. Consequently we conceived s-higraph, an enrichment of higraphs that could. We give an algebraic specification of it as a completion of the one for higraphs.

2. HIGRAPHS AND S-HIGRAPHS

Higraphs are graph-like diagraming objects which were first introduced by Harel and used for a variety of applications. In this paragraph we will give a short description of how higraph are used, that is, of what they actually mean, rather than a mathematical definition for them.

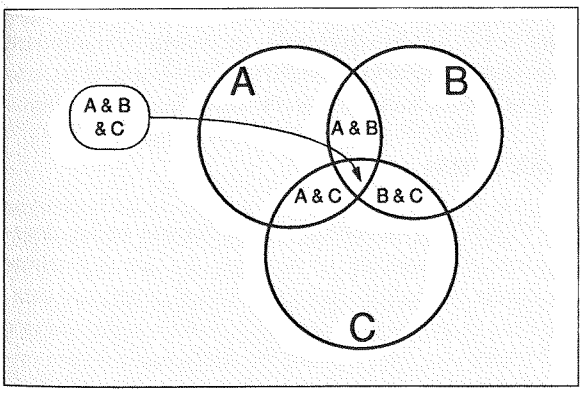


Figure 1: A typical Euler-Venn diagram. The intersection of the circles may be interpreted as conjunction.

A higraph is a set of objects, called blobs, connected by edges . The set of blobs is required to be **finite**; the edges (*hyperedges*) may connect two (or more) blobs at a time; this means that they show a binary (n-ary) relationship among the nodes. In the following we will restrict ourselves to ordinary, i.e. binary, edges.

The basic property of higraphs that makes them different from graphs is the capability of blobs to **intersect** each other. This property, together with the usual representation of blobs as rounded rectangles, makes higraphs look very similar to standard Euler-Venn diagrams used to represent sets and their intersections (figure 1). A blob totally enclosed in another blob is called a *subblob* - clearly resembling the more usual term *subset* - of the enclosing blob, while a blob containing no other blob is said to be *atomic*.
Another way in which higraphs differ from Euler-Venn diagrams is the convention that **every** set of interest be explicitly identified by a suitable closed curve (a rounded rectangle in Harel's notation, which we follow). This convention is mainly aimed at avoiding ambiguities which arise in connection with the use of Euler-Venn diagrams as shown in figure 2 below.
Following the above convention, an area which is not encircled by an atomic blob represents nothing at all1.

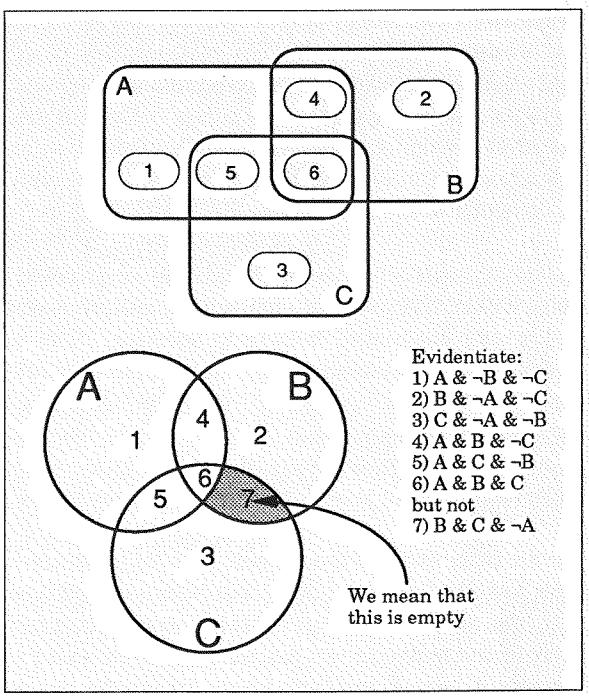


Figure 2: This higraph clearly shows that the set (B & C & -A) is empty by the fact that there is not a corresponding blob, while in the analogous Euler-Venn diagram the ambiguity can not be avoided because it strictly depends on the geometrical position of the contours.

(1) Another feature of higraphs is that a *partitioning function* is defined over blods, thus giving rise to an *unordered cartesian product* but, since as for now we make no use of such a feature, we will omit to describe it (see [Har88]).

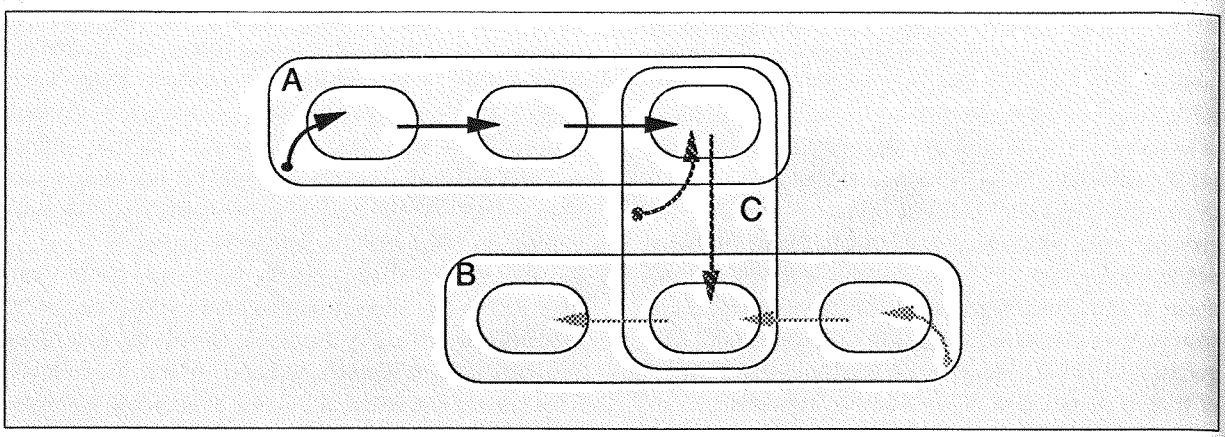


Figure 3: A simple s-higraph. Do note that every blod has its (first generation) subblobs structured in a chain.

We conclude our brief account on higraph by observing that while graphs and hypergraphs are well suited to describe relationships among elements of a given **set**, and so are Euler-Venn diagrams with respect to set-theoretical relationships in a given **collection of sets**, higraphs seem to capture the advantages of **both** representations. In our opinion this shows that there is a crucial advantage in higraph-based formalism, especially in computer science.

In dealing with certain uses of higraphs - e.g. the representation of *documents* - the need of an enrichment to higraphs has arisen. Informally speaking, blobs in a standard higraph are completely identified by their subblobs so that two blobs having exactly the same blobs are actually the same object - just as two sets obtained as unions of the same family of sets are equal **regardless of the order in which the sets are considered**.

On the other hand, if a document is to be represented by a blob then an order does exist within its subblobs (e.g. the order of the pages within a text). Therefore a visual and mathematical feature ought to be added to higraphs in order to represent a total, linear order among the subblobs of a given blob: in other words we must express the fact that such subblobs are actually elements of a **chain**.

However, Harel himself developed higraphs keeping statecharts in mind - that is, objects visualising a process or a finite states machine - and in such context he used arrows starting from nowhere in a given blob and pointing to a subblob of that same blob, thus indicating such subblob as the default state within the state symbolised by the outer blob: the starting state. By making the assumption that (a) in each non-atomic blob there is a *first* subblob and that (b) all other subblobs are connected so as to form a chain (avoiding circularity), we obtain what we have called a **s-higraph**.

It must be noticed that while chains of blobs are normally visualised by means of arrows - and so are edges - the nature of the edges will, in general, be different from that of the links between a subblob and its successor, the latter being precisely an order relationship.

3. RHYTHM

RHYTHM is a hypertext system project under development at University of Bologna. Its aim is to allow the reader to modify, associate and annotate any data. We tried to bypass the distinction between author-creator and reader-user, and enforce the distinction between original version and subsequent alterations instead.

The RHYTHM system is distributed: it is to allow transparent access to all data independently from their position in the network. There are two main modules to obtain this, called *server* and *browser*. The server runs on a multitasking machine and takes care of the loading, preparation and shipment of data; the browser is the only interface of the system with the user and manages all presentation and visualisation of data.

Hypertextual features are guaranteed independently from the data format: the server stores hypertextual data in

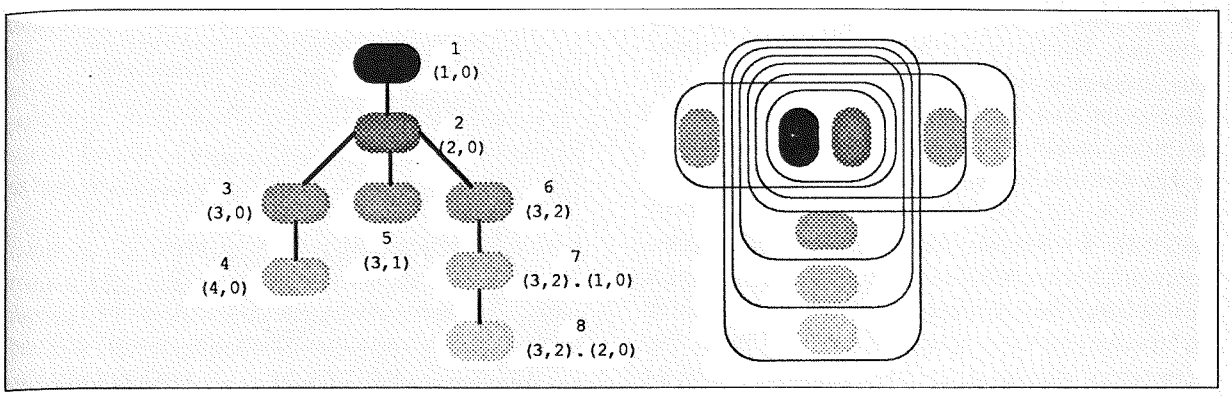


Figure 4: A delta tree of a document and the corresponding higraph: higher numbers gives chronological order of the insertions, and lower numbers give the address of the tree nodes for the system.

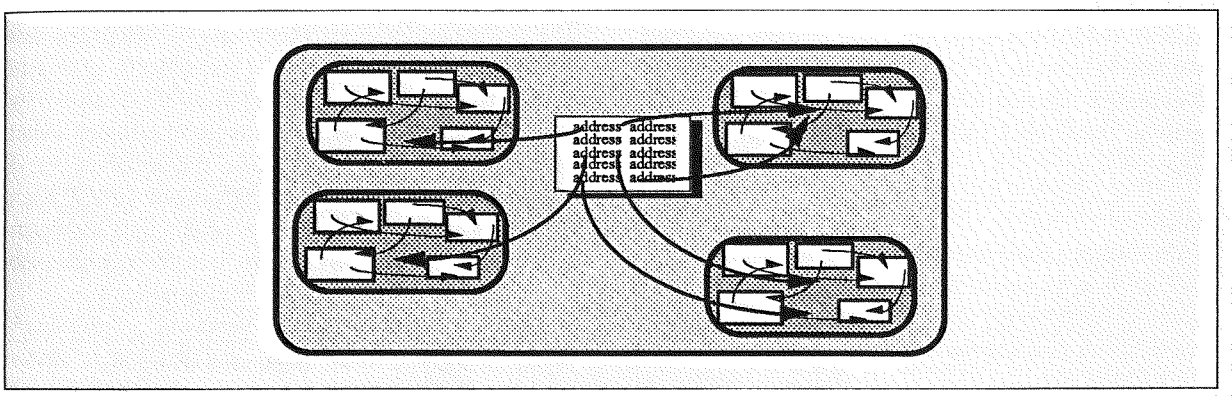


Figure 5: A path introducing some subpaths.

separate files and can accept links and modification independently of the data format; the original data is left intact, and it is the browser's duty to show it at its best. All browsers can show elementary types of data (ASCII texts, bitmaps, etc.), and enhanced browser may be able to accept other data formats, as decided during its development. Special software filters can be activated to convert an unacceptable data format.

We manage full *authoring* and full *versioning* by separately saving the original document and its modifications. By versioning we mean showing the document in all saved versions that contributed to its actual state and by authoring we mean activating together or separating the original version document and all modifications made by others than the first author2. We obtain them through intensive use of *delta versioning*: all modifications made to a document (additions, deletions, moves), i.e. all differences between this and the old version, are saved in a different file, called *delta*; all delta forming the chain (or generally the tree) of modifications to a document are then applied sequentially to the original document to get the desired version.

The RHYTHM system easily manages non-linearity and hierarchies in information structure: strictly hypertextual information (such as links) are saved in a separate document, called *path*. The path document is independent from the data to which it refers; it is possible to have a path connecting many documents, so that a general overview over a certain subject can be given, or to have many paths over the same document, so that

(2) As it can be seen, there is not a big difference between authoring and versioning: they both are modifications to a document, but versioning is said to happen when the modifier is the original author. In this case, the history of the document goes on, and so does its title, all links pointing to it.

it be possible to have many different ways to show the same data. What's more, the path is subject to authoring, versioning and, mainly, it can be origin and destination of links. In this way hierarchic paths of increasing specificity can be created, just like folders or directories in hierarchical file systems.

RHYTHM has a precise structure of its internal documents: all known documents are made of three parts (two of which may be empty): the server script (or *.s*), the browser script (or *.b*) and the raw data (or *.d*). The server script contains all the information needed by the server to find, load and reorganise the content and the links starting from the document.3. The browser script contains all instructions needed to show the data (formatting, margins, font, style, etc.), and the third part contains the data to be shown. The .b and .d parts are of exclusive interest of the browser. The server just needs to understand and manage the .s part.

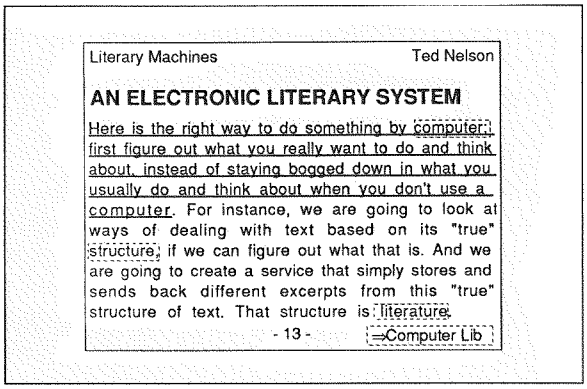


Figure 6: A document for RHYTHM.

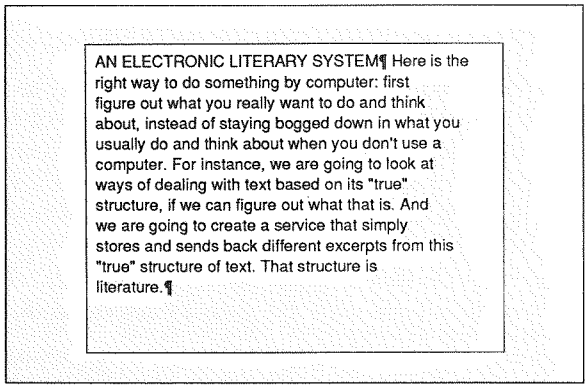


Figure 7: Part .d of the document in fig. 6.

(3) A path is clearly a document made only of the .s part.

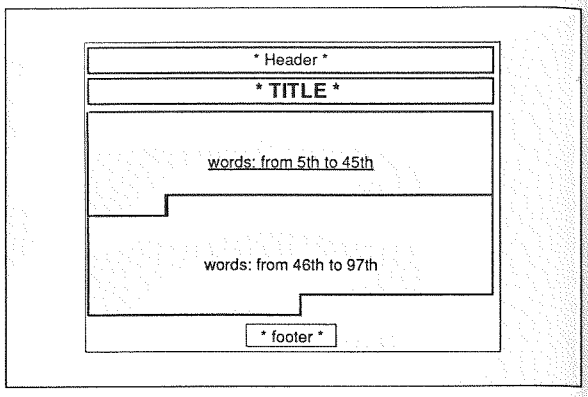


Figure 8: Part .b of the document in fig. 6.

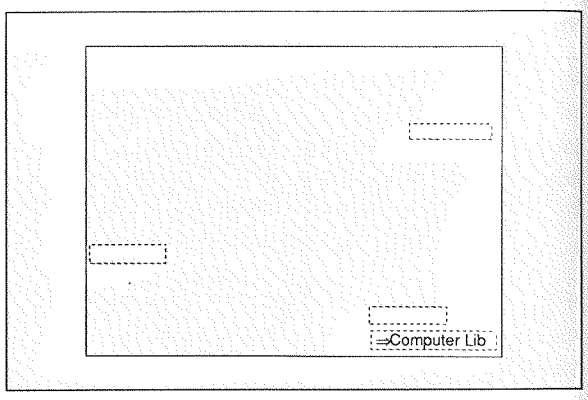


Figure 9: Part .s of the document in fig. 6.

For example, a single line with the file name of the document is the only .s data necessary to add a simple document (ASCII text) to the database: this shows that it should be really easy to introduce generic paper documents in the database, as books or newspaper articles: they just have to be scanned and OCRed.

This three-parts document division helps keeping the three issues relative to data entry disjoint: composition, visualisation and content; modification of one does not alter in any way any of the others. All modifications are made with absolutely no content copying: though it be shown in myriads of different ways, a certain content has to be present on disk just once.

4. HIGRAPHS AND S-HIGRAPHS IN RHYTHM

Higraphs, though they have mainly been sought to manage documents better than graphs do, were found useful also for other problems. In our preliminary specifica-

tions, every document had one owner and was stored in one disk of one machine. This, it has been seen, is severely limiting: a main feature of RHYTHM is to support work group; another is the total distribution of resources so that their use be as elastic and transparent as it is possible.

Later on, the decision of using higraphs let us build a more elastic model for those problems. We try therefore to treat as blobs not only arbitrary chunks of data, but also strongly structured database records, such as information about users or machines.

A group will then be an arbitrary set of users. A group is born after explicit request, or automatically when starting an *argument*(4) among two or more users. This enables us to consider the argument as a single document, which can then be shown by one icon and few links, rather than lots of separate documents (document XXX, Notes to document XXX, Notes to the notes to document XXX, etc.). Edges among nodes may be used to represent free relations among users' groups, such as cooperate or depend, etc.

Analogously, arbitrary sets of machines we call *networks*. A network may effectively represent the physical connection among machines (LAN, WAN, etc), but allows us also to manage *co-existence*: a very common document, for example the main text on a subject, may be stored on many different machines, leaving the system to decide which machine is free enough or comfortable enough to get the data from(5). Edges may represent physical connections, phone lines, administrative dependencies or other things.

5. ALGEBRAIC SPECIFICATION OF HIGRAPHS

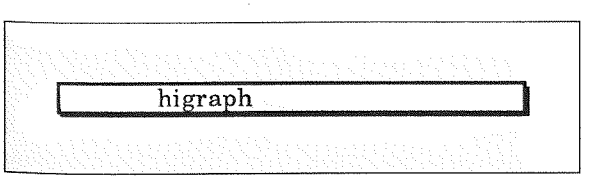


Figure 10: The TYPES section of the A.S. of higraphs.

(4) We say we have an argument when a comment to a document gets an answer by the author of the original document himslef.
(5) The only condition to co-existence is the stability of the document with regard to versioning: that is to ask the document to be stored in the definitive form.

The **TYPES** section

Our algebraic specification (AS) for higraph uses two well known ADT: natural numbers and boolean values with their usual operations; besides, we use the newly introduced type content, about which we will talk later. We also introduce two new sorts - blob and higraph - which are the main objects dealt with in the present AS, and a derived type, string[blob], used to cope with finite sets of blobs of arbitrary (i.e. variable) cardinality.

What we are actually defining is a *higraph with content* and not just a higraph in Harel's sense. In other words, Harel's higraphs are, like graphs, objects of a mainly geometric nature while our use of higraphs involves a remarkable amount of explicit use of their set-theoretical properties. This happens because we are above all interested in knowing what happens to the items (bytes, users, machines) contained in a higraph's blobs when certain modifications of such higraph take place.

We therefore introduced type content, which may be seen as a set algebra with an empty set (EMPTY), a union operation (UNION) and a difference (DIFF); a predicate SUBCONT also exists and represents the set-theoretical inclusion.
Finally, standard operations are assumed to exist for type string[blob], such as NULL, the empty string, CONCAT, the concatena-tion of strings, RESTOF, INSTR and LADD, the adding of an element at the left end of a string; simple properties of strings are assumed to be known, e.g. a string is either empty or of the form LADD(element, string).

The **FUNCTIONS** section

We put some bracketed equations among the functions in the above table; this was aimed at helping to distinguish between definitions of functions solely in terms of other functions, and real, non-tautological equations which are listed in the **AXIOMS** section and which express non-trivial properties of the functions and types we are talking about.

We have two blob-constructor functions: BLOB and BGROUP. BLOB takes a 'piece' of content and returns an atomic blob having precisely that content; BGROUP

```
DELATOM higraph blob → higraph
SUPPRESS higraph blob → higraph

MAKELINK higraph blob blob → higraph
DELLINK higraph blob blob → higraph

SUBBLOB blob → string[blob]
ATOM blob → bool
        [ATOM(b) = if (SUBBLOB(b) = NULL) then True else False]
IN_HIG higraph blob → bool
LINKED higraph blob blob → bool
ARE_LINKED higraph string[blob] string[blob] → bool
NUMBLOB higraph → nat
```

Figure 11: The FUNCTIONS section of the AS of higraphs.

takes an arbitrary set of pairwise disjoint blobs and returns a blob having such blobs as direct (i.e. first generation) subblobs. These two functions reflect the geometrically obvious fact that a blob be either atomic or built by surrounding other blobs with a rounded rectangle.

The accessor functions for blobs are CONTB, SUBBLOB and ATOM; CONTB returns the content of a blob and requires an auxiliary function for its definition, CONTS, returning the overall content of an arbitrary number (string) of blobs; the two functions call each other recursively. This is due to the intrinsically recursive nature of blobs, as stated above: the content of a non-atomic blob has to be ascertained by checking all of its subblobs that, in turn, may be atomic or not,

and so on. SUBBLOB returns a string of blobs: the direct subblobs of its argument. ATOM returns **True** when its argument is an atomic blob.

The only transformer function for blobs is BSEP(6). BSEP takes an atomic blob and part of its content and returns a new blob having exactly two subblobs, one of which contains the part of content used as argument and the other contains what is left of the content of the previous blob; BSEP is actually a mean to isolate a certain portion of content of a blob.

The only constructor function for higraphs is the constant VOID: it represents the trivial (empty) higraph.

Accessor functions for higraphs are CONT, NUMBLOB, IN_HIG, LINKED and ARE_LINKED. CONT is the homologous of CONTB and returns the whole content of a higraph. NUMBLOB returns a natural number expressing of how many blobs its higraph argument is made of. IN_HIG takes a blob and a higraph and checks to see whether the blob belongs to the higraph. LINKED checks the existence of a link starting from a blob and ending in another in a given higraph.

ARE_LINKED performs the same check between each element of a certain string of blobs and each element of another; the need for such a function is the same for having CONTS together with CONTB; in this case too we have a form of mutual recursive calls between LINKED and ARE_LINKED.

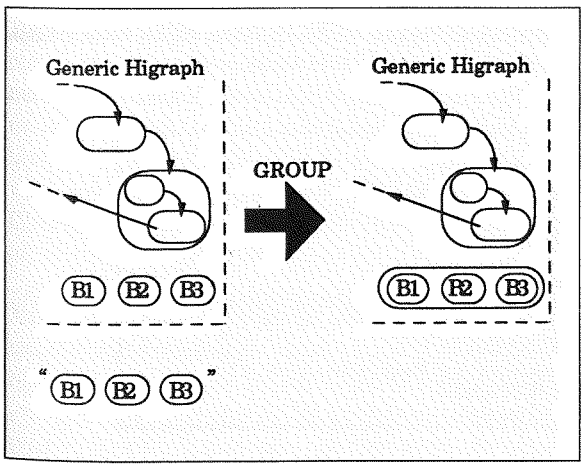


Figure 12: How GROUP works.

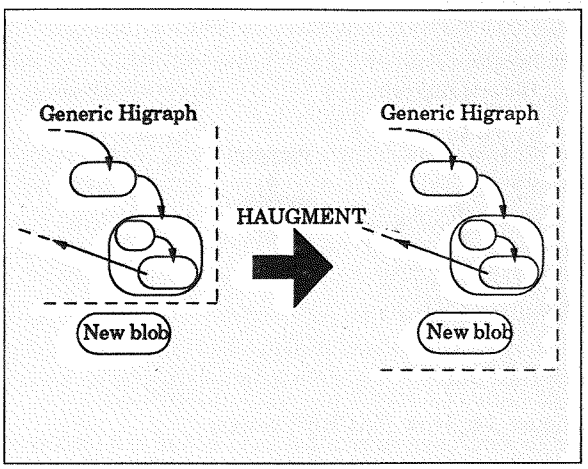


Figure 13: How HAUGMENT works.

The transformer functions for higraphs are HAUGMENT, GROUP, HSUBST, DELATOM, SUPPRESS, MAKELINK and DELLINK. HAUGMENT adds a new blob to a higraph (figure 13); GROUP draws a rounded rectangle around some specified blobs (figure 12); HSUBST substitutes an atomic blob with a new one, possibly non-atomic but having the same content as the former blob (figure 14); DELATOM deletes a specified atomic blob from a higraph (figure 15), while SUPPRESS does the same thing with more general (i.e. non-atomic) blobs; here, again, a two-function approach is required by the recursive nature of blobs.

The above described functions are all that is needed to express the set theoretical relationships among blobs and their content in any higraph, possibly with the aid of blob related functions (BGROUP, CONTB, etc.). Nevertheless it is extremely useful to define functions which perform the most common tasks in the use of higraphs; such functions are precisely those whose definitory equations appear in the functions section: AUGMENT and SEP.

(6) It is technically possible to define BSEP in terms of other functions, and we did, but for the sake of clarity we will often refer to BSEP as a primitive function.

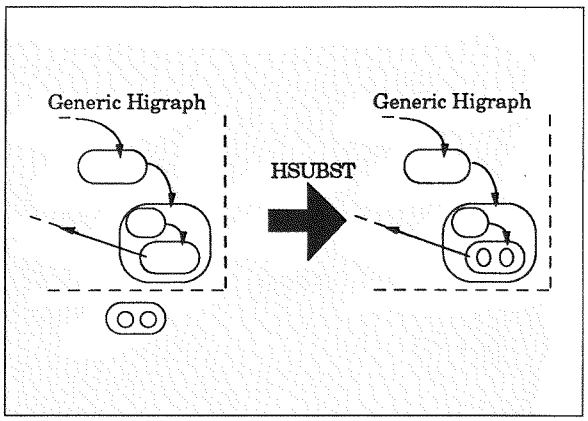


Figure 14: How HSUBST works.

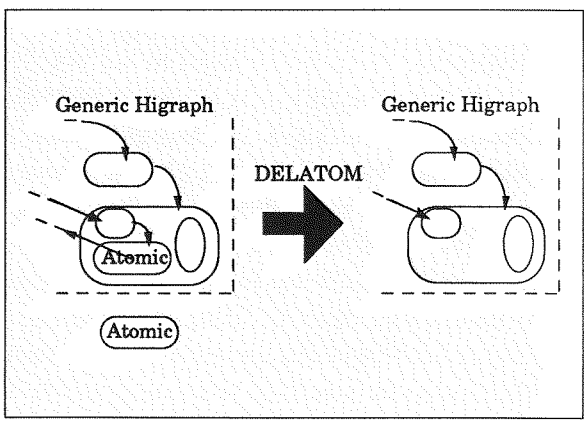


Figure 15: How DELATOM works.

Finally, MAKELINK builds a link from a blob to another (the order is relevant, of course) while DELLINK destroys it.

The **PRECONDITIONS** section

While some of the preconditions in figure 16 are expressed by means of boolean functions, others are not. We did so because writing all preconditions as boolean predicates, though possible and rigourous, would have been exaggerately complicated(7) and unnecessary for the understanding of the subject.

The precondition for BLOB is mainly aimed at avoiding trouble should an attempt be made to refer to a content that does exist in a given higraph but that is shared among disjoint blobs, which means that no blob having exactly the named content exists in the said higraph; besides, the assumed precondition has the advantage of allowing us to postulate some useful properties of BLOB (see the AXIOM section below) that would otherwise be false in the cases excluded by the precondition, and all this is achieved without loss of generality: we mean that every conceivable blob may be built using BLOB and the other blob-valued functions *respecting the given preconditions*. This is also true for all other functions and preconditions we are going to discuss.

The precondition for BGROUP is also due to our will to have BGROUP be the 'inverse' function of SUBBLOB, in the sense that grouping a set of blobs and then asking what the subblobs of the newly built blob are should result in returning the original set (string) of blobs. BSEP requires its argument to be atomic, so that the content to be manipulated is unambiguously de-

(7) The rigorous writing of all preconditions implies the introduction of several supporting functions having no other use.

```
PRECONDITIONS:

pre    BLOB(c)=(content must be new or belonging to an atomic blob)
pre    HAUGMENT(h,b) = (not IN_HIG(h,b))
pre    BGROUP(sb) = (blobs of sb must be pairwise disjoint)
pre    GROUP(h,sb) = (blobs of sb must exist in h)
pre    BSEP(b,c) = (SUBCONT(c,CONT(b)) and ATOM(b))
pre    HSUBST(h,b1,b2)=(IN_HIG(h,b1) and ATOM(b1) and EQUAL(CONT(b1),CONT(b2)))
pre    DELATOM(h,b) = (ATOM(b) and IN_HIG(h,b))
pre    LINKED(h,b1,b2) = (IN_HIG(h,b1) and IN_HIG(h,b2))
pre    ARE_LINKED(h,sb1,sb2) = (strings must be non-empty and their blobs  exist in h)
pre    MAKELINK(h,b1,b2) = (IN_HIG(h,b1) and IN_HIG(h,b2))
pre    DELLINK(h,b1,b2) = (IN_HIG(h,b1) and IN_HIG(h,b2))
```

Figure 16: The PREOCONDITION section of the AS of higraphs.

fined as a whole; on the other hand, a request to isolate a portion of content shared among different subblobs of a certain blob can be handled by repeatedly calling BSEP on the different subblobs. The request that the portion of content to be separated be part of the content of the manipulated blob is obvious but nevertheless necessary.

For the use of HAUGMENT the blob to be added to the higraph must actually be new. HSUBST needs its two blob-arguments to have the same content because this is the very reason of its definition: it has to be a tool for substituting an unstructured (i.e. atomic) bob with another, more richly structured one without altering the higraph's overall content. DELATOM is meant to delete atomic blobs only, since the destruction of structured blobs may be recursively defined on this base (see SUPPRESS below).

All other functions simply require their arguments to exist in the higraph they are being applied to.

The **AXIOMS** section

Axiom (B) expresses the fundamental geometric property of higraphs (especially versus s-higraphs): when grouping blobs, the order is irrelevant; in other words, blobs having the same direct subblobs are equal.

Axioms (C1), (C2) and (C3) should be regarded as a whole; they express the fact that an atomic blob (see also (A1) and (A2)) has the content it was explicitly built to hold by means of BLOB, while the content of a non-atomic blob is the UNION of the contents of its subblobs. The recursion determined by the way blobs are built clearly appears here and explains why CONTS is needed. Note that, since the only true blob-valued functions are BLOB and BGROUP, every time a function with blob-arguments is defined for all arguments of the BLOB(c) and BGROUP(sb) form, it is then defined for *all* blobs.

Axiom (C4) tells us about the effect of the various constructor and transformer functions on the content of a higraph: it actually is the *definition of content of a higraph*. The trivial higraph (VOID) turns out to be empty; HAUGMENT increases the content of the higraph it is applied to by an amount equal to the content of the new blob being added; DELATOM takes away from the higraph's content the content of the blob being deleted; all other functions leave the higraph's content unchanged.

The same task is performed by (N1) with respect to the changes occurring to the number of blobs of a higraph that is being created or modified. The VOID higraph has no blobs; HAUGMENT and GROUP both add one new blob to the higraph; HSUBST enriches the higraph with as many new blobs as the subblobs of the newly added blob are(8). DELATOM obviously causes a decrease of one in the number of blobs of the higraph.

Axioms (A1) and (A2) are self-explaining: blobs of the BLOB(c) form are atomic; all other blobs, which are necessarily of the BGROUP(sb) form, are not.

Axiom (LI) reports about the existence of a link between two given blobs in a certain higraph: it is actually an explicit definition of the LINKED function; is a bit complex and we will only informally describe its meaning; the main concept is that if two blobs are linked we also consider the first blob to be linked to each subblob of the second, as well as each subblob of the first is considered linked to the second; in other words, a link starting from (ending in) a blob is totally equivalent to a set of links starting from (ending in) all subblobs of that same blob and ending in (starting from) the same place as the previous link. Therefore three are the possible causes of the existence of a link between two blobs: it may have been explicitly created by MAKELINK; it may have been generated by grouping certain blobs all of which are already linked with a certain blob, in which case the same link exists for the new blob; it may have been implicitly created by HSUBST, because the introduction of a new blob in the place of an old one implies the existence of a link between (any subblob of) the new blob and any other blob with which the substituted blob was previously linked.

Axiom (S1) expresses the key property of atomic blobs: they have no subblobs. Axiom (S2) that of non-atomic blobs: they may be thought as having been built by grou-

(8) A much more frequently used function, defined after HSUBST, is SEP. It implies the creation of *two* new blods: the one with the content one wanted to isolate and the one containing the rest of the content of the manipulated blob.

```
LI    LINKED(h,b1,b2) = case
          (h = MAKELINK(h1,b1,b2))   :: True
          (h = DELLINK(h1,b1,b2))    :: False
          (h = HAUGMENT(h1,b))       :: if ((b1=b) or (b2=b))     then False
                                                                  else LINKED(h1,b1,b2)
          (h = GROUP(h1,sb))  :: case
              (b1=BGROUP(sb))        :: ARE_LINKED(h,sb,"b2")
              (b2=BGROUP(sb))        :: ARE_LINKED(h,"b1",sb)
              (else)                 :: LINKED(h1,b1,b2)
          (h = HSUBST(h1,b,b3))      :: case
              (b1=b3) :: case
                    (b2=b3)                          :: LINKED(h1,b,b)
                    (INSTR(b2,SUBBLOB(b3)))          :: LINKED(h1,b,b)
                    (else)                           :: LINKED(h1,b,b2)
              (b2=b3) ::                        if INSTR(b1,SUBBLOB(b3)
                                                      then LINKED(h1,b,b)
                                                      else LINKED(h1,b1,b)
              (INSTR(b1,SUBBLOB(b3))):: if INSTR(b2,SUBBLOB(b3))
                                                      then LINKED(h1,b,b)
                                                      else LINKED(h1,b,b2)
              (else)                            LINKED(h1,b1,b2)
          (h = DELATOM(h1,b))        :: LINKED(h1,b1,b2)

IH    IN_HIG(h,b) = case
          (h = VOID)                 :: False
          (h = HAUGMENT(h1,b1))      :: if (b=b1) then True else IN_HIG(h1,b)
          (h = GROUP(h1,sb))         :: if (b=BGROUP(sb)) then True  else IN_HIG(h1,b)
          (h = HSUBST(h1,b1,b2))     :: case
              (b=b2)  :: True
              (b=b1)  :: False
              (else)  :: IN_HIG(h1,b)
          (h = DELATOM(h1,b1))       :: if (b=b1) then False else IN_HIG(h1,b)

S1    SUBBLOB(BLOB(c)) = NULL
S2    SUBBLOB(BGROUP(sb)) = sb

A3    ARE_LINKED(h,LADD(b1,s1),LADD(b2,s2)) =
          if s1 = NULL then
              if s2=NULL     then LINKED(h,b1,b2)
                             else LINKED(h,b1,b2) and ARE_LINKED(h,"b1",s2)
          else if s2=NULL    then LINKED(h,b1,b2) and ARE_LINKED(h,s1,"b2")
                             else LINKED(h,b1,b2) and ARE_LINKED(h,"b1",s2) and
                             ARE_LINKED(h,s1,"b2")

S3    SUPPRESS(h,b) = if (ATOM(b)) then DELATOM(h,b) else
                      SUPPRESS( SUPPRESS(h,FIRST(SUBBLOB(b)), b)
```

Figure 17: The AXIOMS section of the AS of higraphs.

ping their direct subblobs.
Axiom (IH) tells us that no blob belongs to the VOID higraph and that there are only three ways for a blob to have become part of a higraph: it may have been added to the higraph by HAUGMENT, it may have been generated by GROUP in order to surround other blobs or it may have been brought in by HSUBST (either explicitly or as subblob of the introduced blob) in substituting a blob with another; a blob may disappear only by means of DELATOM (or its repeated use in the case of SUPPRESS); MAKELINK and DELLINK have no effect in this respect.

Finally, axiom (S3) explains how the destruction of any blob may be performed by first destroying all of its atomic subblobs and finally destroying the blob which has in turn become atomic (and empty).

6. ALGEBRAIC SPECIFICATION OF S-HIGRAPHS

The main property we want s-higraphs to have is that the building of a blob from other blobs (performed by a function homologous of GROUP) be anti-commutative, an order having been imposed on the subblobs of the newly built blob. In our AS, such crucial difference between s-higraphs and higraphs appears in that the former always posses labeled links, usually of at least two different kinds, while the latter are not required to; this allows us to define a function SGROUP (the homologous of GROUP) which uses one specific type of labeled link to construct a chain of blobs while grouping them. The chain of the subblobs of any blob is thus made of actual links, and these links can be told from the usual ones thank to their special label.

Of course, the additional feature of labeled links may be easily added to higraphs as well and probably will in most applications.

The TYPES section is therefore modified as follows:

s-higraph =	**nat + bool + content +**
TYPES:	blob string[blob] s-higraph label

Figure 18: The TYPES section of the AS of S-higraphs.

The **FUNCTIONS** section is almost identical to that of higraphs with only a few, though fundamental, changes; we therefore list these changes only.

The first change is that each function of the previous AS having higraph-type arguments or value is substituted by a similar function having s_higraph-arguments (value) instead. All such functions have a name obtained by that of their old (higraph) homologous functions by adding a 'S_' as a prefix to remind the reader of s-higraphs. Moreover, as a second change, the three functions dealing with links have to have new signatures (*and* new names, of course) to take into account the existence of several link types: S_MAKELINK and S_DELLINK are to make and destroy links of specified type, while S_IS_LINK has to ascertain the existence of links bearing a specific label.

One exception to the above established notation is that function GROUP becomes OLD_GROUP.

A new important function, S_GROUP, is introduced: it (not OLD_GROUP) is the homologous of GROUP. MAKECHAIN is also a new function and is used in strict connection with S_GROUP: it represents the building of a sequence of links of the CHAIN kind orderly connecting all blobs in a given string so as to form a chain.

The last change is that one constant for type label is introduced, in order to ensure the existence of the 'order links' we want: CHAIN.

All functions that do not have type higraph in their signature are transferred unchanged to the new AS.

The preconditions section undergoes no relevant change: its only modification is that the new signature has to be used in writing the preconditions. Besides, a simple precondition is required for MAKECHAIN.

The modification of the AXIOM section, apart from the new signature being used, lies in the first equation mainly but is of outmost importance. Another axiom, (M), is also added in order to define the new MAKECHAIN function.

Axiom (B) of higraphs is substituted by axiom (SG) as

follows:

```
FUNCTIONS:
        CHAIN → label
        S_MAKELINK s-higraph blob blob label → s-higraph
        S_DELLINK s-higraph blob blob label → s-higraph
        S_IS_LINK s-higraph blob blob label → boolean
        S_GROUP s-higraph string[blob] → s-higraph
        MAKECHAIN  s-higraph string[blob] → s-higraph
```

Figure 19: The FUNCTION section of the AS of S-higraphs.

```
PRECONDITIONS:

pre     MAKECHAIN(h,sb)=(not sb=NULL)
```

Figure 20: The PRECONDITIONS section of the AS of S-higraphs.

```
AXIOMS:

SG      S_GROUP(h,LADD(b,sb)) = MAKECHAIN(OLD_GROUP(h,sb),sb)

M       MAKECHAIN(h,LADD(b,sb))=if (LENGTH(sb)=1)
            then S_MAKELINK(h,b,FIRST(sb),CHAIN)
            else MAKECHAIN(S_MAKELINK(h,b,FIRST(sb)),sb, CHAIN )
```

Figure 21: The AXIOMS section of the AS of S-higraphs.

It should be clear that axiom (B) of the previous specification is false with respect to s-higraphs. Axiom (M) simply states the properties of MAKECHAIN already mentioned above.

7. CONCLUSIONS AND FUTURE WORK

We think that RHYTHM addresses real problems within the hypertext publishing field. We hope that this overview has shown how higraphs can help solve them in an elegant way, though all matters about efficiency have (intentionally) not been taken into account.

Higraphs are a powerful tool for hypertext and lots of other problems, and ADT have shown to really help implementors in getting things straight with less pain and more speed.

Lots of things have been left out, clearly. We will consider some of them in future works; we will mainly focus on the specification of typed links and will get deep into s-higraphs.

ACKNOWLEDGEMENTS

We wish to thank Prof. Cesare Maioli of Program in Computer Science and Prof. Massimo Ferri of Department of Mathematics, University of Bologna, for their substantial help, support and ideas; many thanks also to Dr. Vita Lia Micelli, Dr. Alessandro Amoroso and Dr. Renzo Davoli for their help, ideas and chatter, though we ourselves assume on the responsibility for all inconsistencies we could possibly have put in.

8. REFERENCES

[1] Berge C., GRAPH AND HYPERGRAPHS, NORTH-HOLLAND, 1973

[2] Bosello C.A., Vitali F., HIGRAPHS AS A MODEL FOR HYPERTEXT PUBLISHING, Quaderno n°19, Laboratorio del Corso di Laurea in Scienze dell'Informazione, Bologna, 1990

[3] Conklin J., HYPERTEXT: AN INTRODUCTION AND SURVEYS, Computer, Sept '87

[4] Ehrig H., Mahr B., FUNDAMENTALS OF ALGEBRAIC SPECIFICATION, Springer Verlag, 1985

[5] Guttag J., ABSTRACT DATA TYPES AND THE DEVELOPMENT OF DATA STRUCTURE, Communications of the ACM, June '77

[6] Halasz F.G., REFLECTION ON NOTECARDS: SEVEN ISSUES FOR THE NEXT GENERATION OF HYPERMEDIA SYSTEMS, Communications of the ACM, July '88

[7] Harel D., A VISUAL FORMALISM, Journal of the ACM, May '88

[8] Liskov B., Guttag J., ABSTRACTION AND SPECIFICATION IN PROGRAM DEVELOPMENT, The MIT Press, 1987

[9] Meyer B., OBJECT ORIENTED SOFTWARE CONSTRUCTION, Prentice Hall, 1988

[10] Micelli V.L., Maioli C., GRAFI E HIGRAPH COME STRUTTURE DI BASE DI SISTEMI IPERTESTUALI, Quaderno n°11, Laboratorio del Corso di Laurea in Scienze dell'Informazione, Bologna, 1989

[11] Nelson T., COMPUTER LIB / DREAM MACHINES, Microsoft Press, 1974

[12] Nelson T., LITERARY MACHINES, Project Xanadu, 1981

[13] Nelson T., MANAGING IMMENSE STORAGE, Byte, January 1988

[14] Van Dahlen D., LOGIC AND STRUCTURE, Springer Verlag, 1980

[15] Vitali F., Maioli C., Micelli V.L., RHYTHM: UN SISTEMA DI PUBLISHING IPERTESTUALE BASATO SUGLI HIGRAPH, AICATEXT '89 Proc., AICA, 1989

HYPERNET

Marco Bondi, Maurice Pagnucco
Dipartimento di Scienze dell'Informazione
Hypertext User Group
Università degli Studi
Milano

RIASSUNTO

HYPERNET è un sistema multimediale, basato sul formalismo delle Reti di Petri, che può essere utilizzato per la rappresentazione, l'analisi ed il design di sistemi (dalla specificazione di sistemi industriali alla ingegneria del software) descrivendone le attività. La multimedialità permette un più alto potere di specificazione rispetto a quello che deve essere mostrato quando il modello del sistema viene esplorato e consente all'utente di applicare le tecniche di analisi sviluppate per verificarne le proprietà. Il sistema osservato è organizzato gerarchicamente su più livelli e le diverse informazioni (testi, grafici, suoni) sono associate ai nodi della rete in base al loro significato; i connettori della rete permettono di correlare queste informazioni, utilizzando precise relazioni che risultano dal modello del sistema che l'utente sta analizzando.

ABSTRACT

HYPERNET is a multimedia system, based on Petri Net formalism, which can be used for the representation, analysis and design of systems (e.g. from the specification of industrial systems to software engineering) exhibiting activities within them. The multimedia system permits a more powerful specification of what is to be displayed when the model of a system is browsed, and allows the user to apply developed analysis techniques to verify properties of the system. The observed system is organized hierarchically on several levels and the different information (text, graphics, sound) is associated to the nodes of the net according to the meaning. The connectors allow to correlate this information through precise relationships which result from the model of the system that the user is analysing.

1. INTRODUCTION

HYPERNET is a multimedia system which utilizes the Petri Net formalism for the modelling of system. Petri nets, first introduced by C.A. Petri [18], are essentially designed for representing and analysing systems. HYPERNET possesses several important features which make it particulary suitable for system design in a wide range of fields (eg. applicable to building, office automation, communication protocols, distributed databases, etc.). Some of these features include the ability to describe systems at various levels of detail, their graphical and precise nature and their structural generality.
The multimedia system is essentially concerned with three diverse aspects: the realization of the graphical environment (Net Editor), the analysis of the properties of the net (Net Analyser) and the information to associate to the nodes (Hypertextual Features).
The Net Editor has a number of functionalities that permit the creation of syntactically correct nets (verification of node connections, and parse of arc annotations to confirm their validity) with various levels of detail (net morphisms) and with a variety of styles and forms for the graphical objects: nodes and connectors, according to the requirements. Moreover, it realizes a com-

plete user interface for the creation, selection, movement and manipulation of graphical objects, providing a simple scheme of dialogs necessary for the acquisition of data to associate to the nodes and connectors.
For the analysis of the net properties, the Net Analyser provides the following options : (for Condition Event-nets only) concession, firing, step, contact, conflict, reachability, case class of a net, reachability graph, enlogic structure, liveness, cyclicity and safeness; (for Predicate Transition-nets only) transformation matrix, and; (for both) pre-set and post-set, simplicity, purity, incidence matrix, name S-invariants and name T-invariants.
The Hypertextual environment provides a series of features that permit the user to associate, to the nodes of the net, information (text, graphics, sound) that may be organised hierarchically on several levels and whose contents may be inserted using the associated Net Editor or obtained directly from a text or graphics file created by any Macintosh application (MicroSoft Word™, SuperPaint™, Mathematica™, MathType™, etc.). The images and text may also be obtained using a scanner or similar device.
The information associated to the nodes of the net are related to connectors. They identify towards which node and in which direction the flow of information moves. The mechanisms which allow the user to enter the hypertextual structure are typical mechanisms utilized to the hypertextual models: the double click and the items of the menu as "Child", "Parent". These mechanisms operate on any node of the net and allow the user to verify if a node exists and allow the user to associate to the node the same textual/graphic information or a subnet. If the same textual/graphic information exists, then the tool shows if there is a subnet then the tool allows the user to continue the "browsing" selecting a new node or verifying properties of the net or of the current subnet.

2. HYPERNET - THE APPLICATION

In this chapter a description of the features made available, to the user, by HYPERNET is given. To better understand the options provided by the application we begin with a brief description of the theory underlying Petri Nets. Readers who desire a more thorough explanation of net theory can consult the Appendix or the texts by Best [1] and Genrich [11] .

2.1 Basic Theory

A Petri Net is a triple, $N=<S,T,F>$ in which $S=\{s_1,....,s_n\}$ is a finite set of places with $n\geq 0$, $T=\{t_1,....,t_m\}$ is a finite set of transitions with $m\geq 0$ e $ST =$, and F $(S \times T) (T \times S)$ is the flow relation, a mapping representing arcs between places and transitions.
The arcs represented by F prescribe pre- and post-relations for places and transitions. The set of places that are incident on a transition t is termed the preset of t, and is denoted by $^\circ t = \{ s \mid (s,t) F \}$. The set of places that follow a transition t is termed the postset of t, and is denoted by $t^\circ = \{ s \mid (t,s) F \}$. The preset and postset for a place s are defined similary as the sets of transitions incident on s and following s, respectively.
A marking M of a Petri net $N=<S,T,F>$ is a function M $M: S \longrightarrow \{ 0, 1 \}$, mapping each place in the net into a nonnegative integer. Each integer in a marking indicates the number of tokens residing in the corrisponding place.

2.2 The Net Editor

The graphical functionality of HYPERNET permits the user to create graphically complex nets. In fact, these allow the user to allign the graphical objects to any determined angle of inclination (horizontal, vertical, oblique, etc.), to spread the nodes proportionally, to drag the nodes without the need to redesign the connectors and to set default values for the creation of nodes and connectors using any type of style, shape and colour. Moreover, they allow the user to "zoom" the pages from 400% to 25% of their standard dimensions, to curve a connector (selecting the angle of curvature), to effect the update of the current page and to load or construct, dynamically, "palette" pages.
Acquistion of the data that is to be associated to the nodes and connectors of the net is effected through dialog boxes that appear on the screen. The user has the option of entering this data while constructing the object or he can choose to allow HYPERNET to automatically generate a label for the object. At any moment, the user can modify the labeling of an object of the diagram by simply selecting it.
To each object of the net (CE-net or PrT-net) we can associate diverse types of information. Every object must be assigned a name. This is always perfomed implicitly (ie. assigned an internal identifier) by HY-

PERNET and may also be specified explicitly by the user.
For CE-nets, while events and arcs may only have a name, the conditions may also have a mark. If a mark exists in a given condition then it will be shown graphically. This mark may be assigned interactively by the user by selecting the appropriate condition with the mouse or by entering it in the condition's dialog.
However, if the user is using the PrT-net formalism, then he may associate a formula to each transition, a predicate to each predicate and a symbolic sum of tuples to each arc of the net. Before any calculation is performed on the net each arc must be assigned a tuple, otherwise the calculation cannot proceed. As the validity of most results are based on the correctness of the arc labels, HYPERNET also performs a syntax check of all arc labels. If this parse is unsuccessful then the application cannot proceed with any calculation.
The Net Editor permits the user to generalise the net on many hierarchical levels (morphisms). This generalistion may be realised in two different ways: selecting the group of nodes that are to be placed on a lower level (on a new page) and substituting them with a node of the appropriate type or selecting one node and constructing an empty lower level which the user may define later on. The morphism operation is performed only if the group of nodes selected respect the definition of morphism (see previous section).
In the eventuality that the user needs to duplicate part of the net (eg. to save time) he may do so, utilizing copy and paste, and the application will ask if he wishes to maintain the data structure (ie. labels and other associated data), of the duplicate part, or modify it. If the user decides to modify the data structure, this may be performed either automatically by the application or manually through dialogs.
To make the tool more flexible, HYPERNET allows the user to import nets designed with MetaSoftware's MetaDesign™. This feature automatically controls the syntax of the net, associating to each graphical object the suitable data structure. In the case that an object of the net cannot be classified, the application asks the user to supply additional information.

2.3 The Net Analyser

The properties that are analysed by the Net Analyser are essentially of two types: static and dynamic. Static properties regard the structure of the net (graphical representation of the connections and adjacency of the nodes) and are applicable to both CE-nets and PrT-nets. Dynamic properties are related to the evolution of the tokens (PrT-nets) and marks (CE-nets) in the net.
To make the tool more interactive, the selection of the graphical objects, on which the properties are verified, is performed using the mouse and the results are displayed in the status bar (window lying directly beneath the menubar) and/or evidenced graphically using a particular graphical feature. The Net Analyser verifies a property by specifying, as input, a single node or a group of nodes and responds, in ouput, graphically and/or textually. If nothing is selected as input then the property is verified using every object (of the appropriate type) of the net one by one.
The static properties analyzed by HYPERNET consist of: pre-set, post-set, purity, simplicity, incidence matrix, name S-invariants and name T-invariants. These properties will now be discussed with the use of examples.
The pre-set and post-set of the selected node are returned in the form of a string in the status bar and evidenced graphically on the net.
The purity of the net is acknowledged in the status bar and if the net is not pure the offending condition/event (predicate/transition) pair(s) is highlighted on the net.
The simplicity of the net is also acknowledged in the status bar but, in this case, the offending pair(s) of conditions (predicates) and/or events (transitions) are evidenced graphically.
To give an example of invariant calculation we utilize a net which models part of the law n. 898 (1970): "I casi di scioglimento del matrimonio in Italia" (Circumstances in the dissolution of a marriage in Italy) [20].
The incidence matrix of the net is output, on a new page, in the form of a matrix indexed by the names of the conditions (predicates) and events (transitions) of the net. For a CE-net the entries consist of 0's or 1's but for a PrT-net the entries are either 0 or symbolic sums of tuples.
The system invariants are calculated by determining a basis for the relevant homogeneous system of equations ($\mathbf{C}^T.\boldsymbol{i} = \mathbf{0}$ for S-invariants and $\mathbf{C}.\boldsymbol{i} = \mathbf{0}$ for T-invariants, where $\mathbf{C}$ represents the incidence matrix of the net and $\mathbf{C}^T$ its transpose). In fact the invariants generated are calculated using the ring product and are known as name invariants (see [13] - especially for PrT-nets).

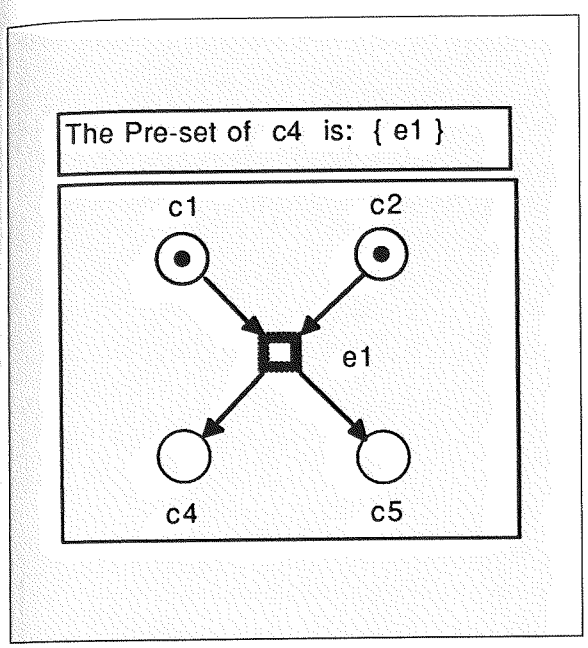


Figure 1: The highlighted event shows the Pre-set of the selected condition

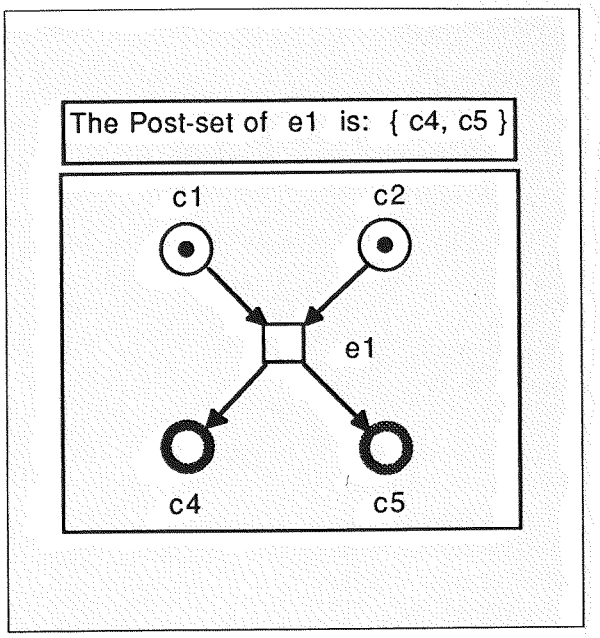


Figure 2: The highlighted conditions show the Post-set of the selected event

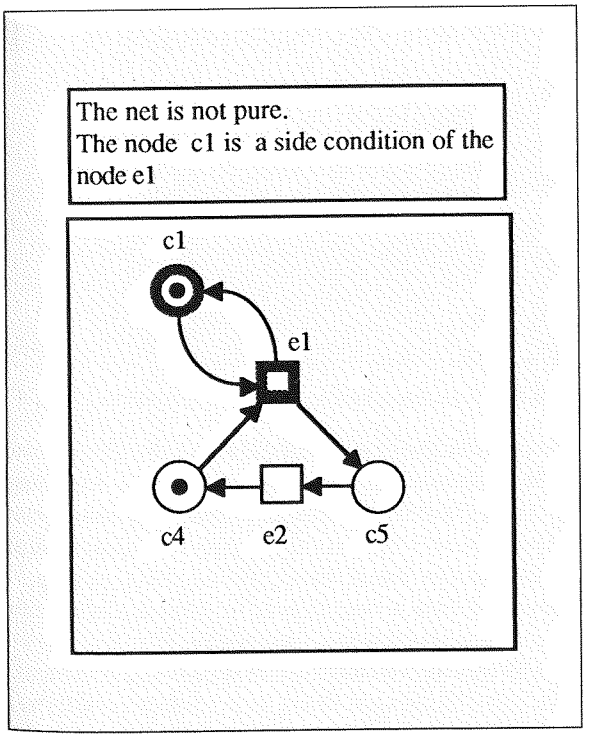


Figure 3: The couple of nodes which let the net be non-pure

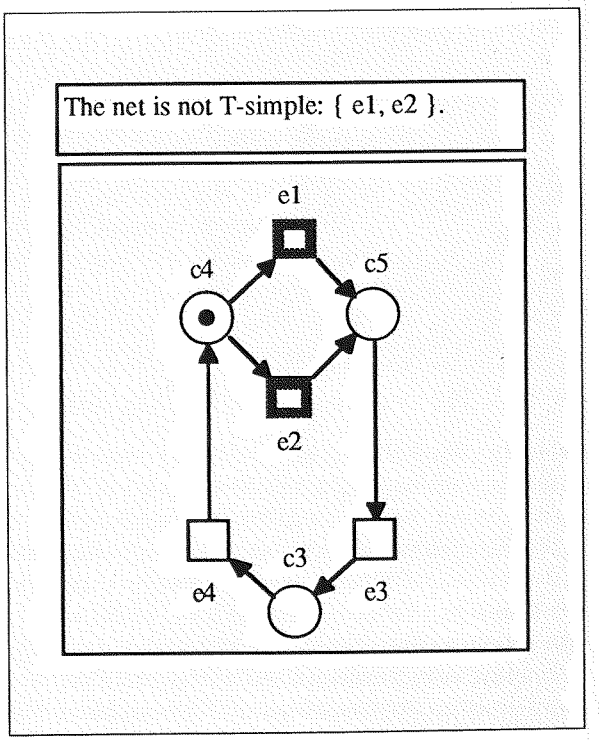


Figure 4: The couple of events which let the net be non-simple

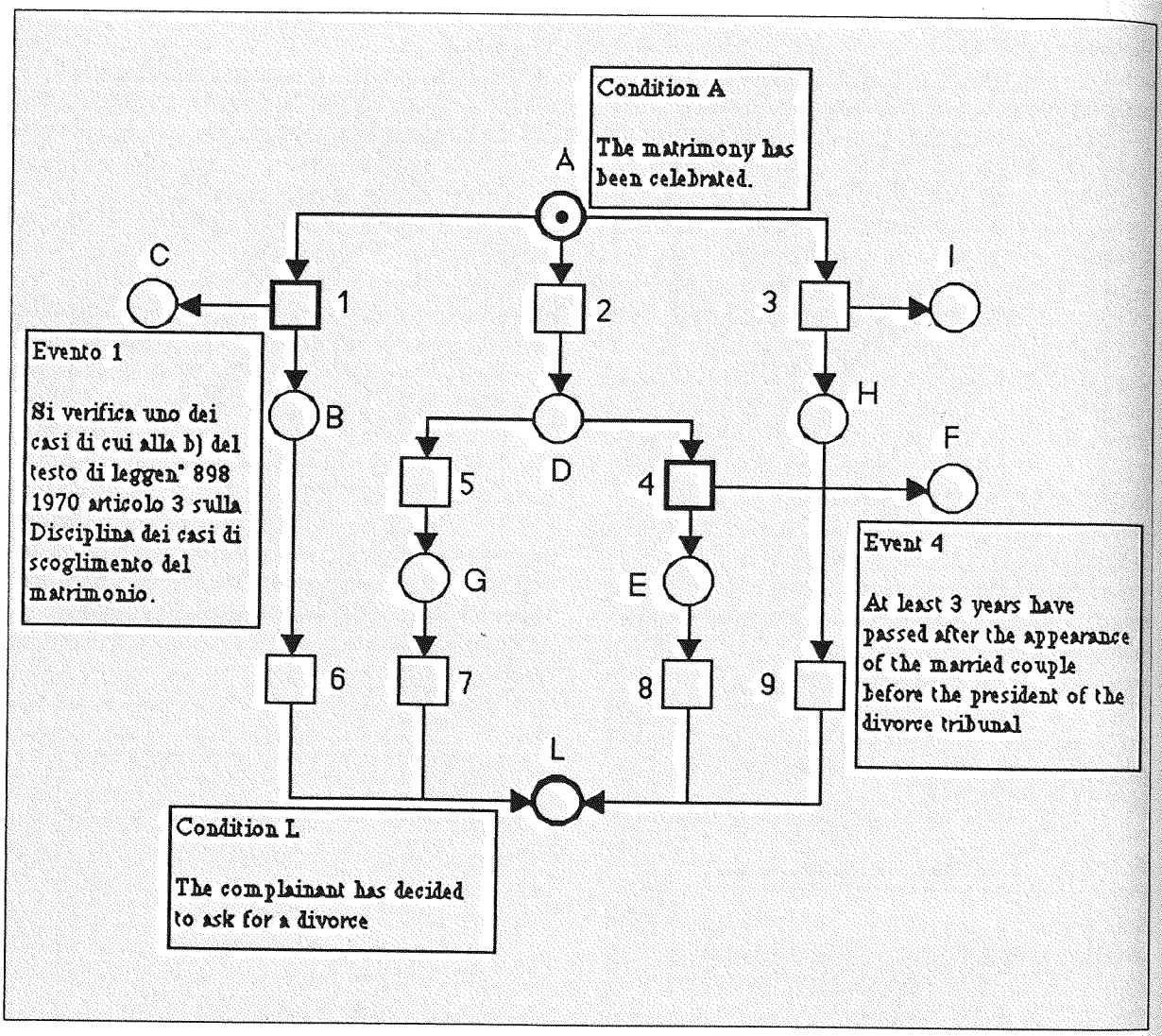


Figure 5: The net that models the third article of law n° 898 - 1970 "Casi di scioglimento del matrimonio in Italia"

Moreover, for PrT-nets, the resulting invariants may contain symbolic components and, as a result, further manipulations and calculations (eg. projection [10) may be applied to transform these invariant vectors into a form where all components are integer quantities (thus giving a merely quantitative view of the net).
The mathematical calculations involved are actually performed using Mathematica™. Any invariant vectors derived are displayed in matrix form and the user is also given the option of seeing them highlighted, graphically, on the net and written in the status bar in the form of an equation.

The dynamic properties that the user may choose from is dependent on the formalism adopted for designing the net: CE-net or PrT-net formalism. The analysis of the properties of CE-nets is based on the calculation of the reachability graph while the analysis of the properties of PrT-nets is based on the calculation of certain system invariants as described above.
For CE-nets the user has the following choices at his disposition: concession, firing, step, contact, conflict, reachability, case class of a net, reachability graph, enlogic structure, liveness, cyclicity and safeness.
Concession, contact and conflict of the selected event(s)

Incidence Matrix
(10 Predicates x 9 Transitions)

Pr \T	5	2	3	6	7	8	9	1	4
G	<>	0	0	0	- <>	0	0	0	0
B	0	0	0	- <>	0	0	0	<>	0
D	- <>	<>	0	0	0	0	0	0	- <>
H	0	0	<>	0	0	0	- <>	0	0
I	0	0	<>	0	0	0	0	0	0
C	0	0	0	0	0	0	0	<>	0
E	0	0	0	0	0	- <>	0	0	<>
F	0	0	0	0	0	0	0	0	<>
L	0	0	0	<>	<>	<>	<>	0	0
A	0	-<>	- <>	0	0	0	0	-<>	0

S-Invariant

Pr \i	i1
G	1
B	1
D	1
H	1
I	0
C	0
E	1
F	0
L	1
A	1

Figure 6: This figure shows the incidence matrix and an S-invariant

are confirmed (or not) in the status bar in the form of a message. Also, in the case where contact (conflict) exists the offending conditions (events) are evidenced graphically.

The operations of "firing" and "step" are realized graphically by moving the tokens from the conditions in the pre-set of the selected events (one event only in the case of "firing") to the conditions of the relevant post-set.

The case class of the net is listed in the status bar while the reachability graph is designed on a new page in the form of a graph whose nodes are the cases of the net and whose arcs represent the firing of an event.

The operation of "reachability" may be effected in three different ways (each taking the current marking as the initial case): the user may specify a list of events and a case to reach and HYPERNET will display a message in the status bar after having verified if the case is reachable or not with this set of events; the user may specify a list of events and HYPERNET will display the final case, or; the user may specify the case to reach and HYPERNET will display the list of events that permits it to be reached.

The objects that belong to the enlogic structure of the

net (fact, violation and process) may be inserted into the net using the Net Editor and then verified. In the case where this verification fails (the object does not belong to the enlogic structure) the object is deleted from the net.
Liveness, cyclicity and safeness are confirmed (or not) by a message in the status bar. In the case that one of these properties is not verified, the offending event(s) is highlighted. For PrT nets we may also obtain the transformation matrix of the net. This matrix gives the user a totally quantitative view of the net as all its entries are integer quantities.

2.4 Hypertext Features

An HYPERNET document consists of one or more pages in which graphical objects (nodes and arcs) and text are created. Each page corresponds to a window that has all the characteristics of a standard Macintosh window. Pages and objects may be linked hierarchically (morphisms and graphical/textual information) or not (nets that belong to the same document but that specify systems, or parts of a system, that do not interact with each other).
To navigate within this hierarchical structure the user may utilize diverse functionalities. The functions "Page Structure", "Child" and "Parent", selectable directly from the menu, permit the user to move within the hierarchical structure of the document by selecting the sub-net or by selecting a node that has an hierarchical structure associated to it. The "double-click" on a node and the "click" in the "close-box" of a window, allow the user to pass from one level of the document hierarchy to another (if it exists) or to evidence the graphical/textual structure (if any) associated to the node.
In the above figure, the user can choose the hierarchical level in which to enter by clicking on the name of the relevant page.
The previous figure displays a morphism that has two hierarchical levels in which the second is dependent on the first (in particular, on event e1). To pass from one level to the other it is sufficient to "double-click" on the event, e1.
To increase the descriptive and modeling power of Petri Nets, HYPERNET allows the user to associate, to the graphical objects of the net, information that is graphical and/or textual in nature. This graphical/textual information may be imported from any Macintosh application by using the Scrapbook and/or Clipboard.
Using the standard Macintosh operations of Copy and Paste information may be transferred from an application to the net via the Scrapbook (or Clipboard). The user may then select, with the mouse, a node with which this information is to be associated. Our experience has shown that this type of operation is most easily performed using the Macintosh in Multifinder mode (to switch between applications) and utilizing the Clipboard. Another method that may be used to associate textual information (only) to objects of the net is to digitize it directly by using the Net Editor.
To view the information associated to an object of the net it is sufficient for the user to position the pointer of the mouse on the object and "double-click". Any graphical/textual information associated to the object, and not visible to the user, will appear. The same operation can be used to hide any visible graphical/textual information that is associated with an object. A piece of graphical/textual information can also be made to d(ppear by "double-clicking" inside it.
The objects of the net to which graphical/textual information has been associated are automatically highlighted by HYPERNET. This allows the user to quickly and easily see where information may be contained.

3. THE DEVELOPMENT ENVIRONMENT

HYPERNET has been developed on an Apple Macintosh II utilizing MetaSoftware Corporation's Design/OA™ [4] environment with THINK's LightspeedC™ compiler and Wolfram Research's Mathematica™.
Design/OA™ provides the user with procedures that allow him to proceed rapidly toward the realization of an application, concentrating his attention on high-level problems, directly related to the application, rather than on low-level problems concerning the graphical interface. In this way it has been possible to associate, to the graphical and hypertextual flexibility of Design/OA™, a series of procedures that add, to Design/OA™, the analysis a voriety of Petri Nets properties.
Mathematica™ [24] is a powerful application which allows the user to perform both numerical , symbolic and graphical calculations as well as providing a high-level programming language for defining new functions. In order to save time while developing HYPERNET we have created an interface between our program, written using Design/OA™, and a small Mathe-

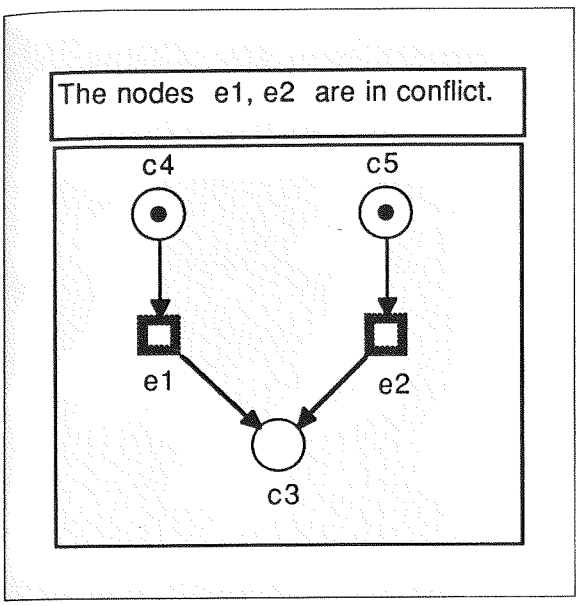


Figure 7: The highlighted events are in conflict

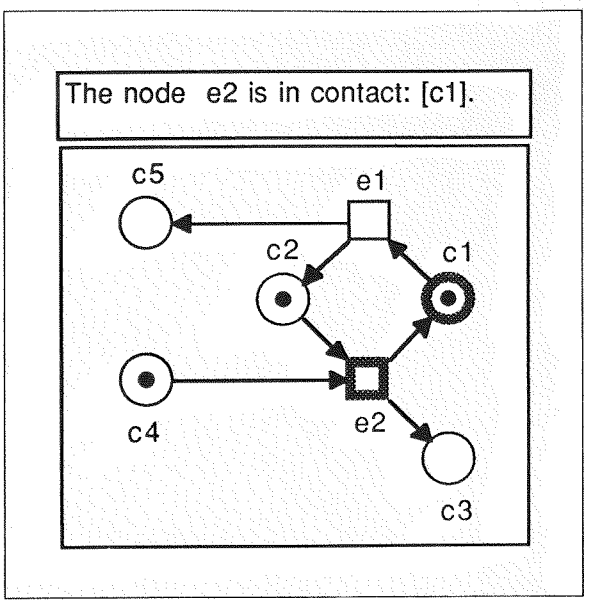


Figure 8: The highlighted selected event is in contact with the highlighted condition

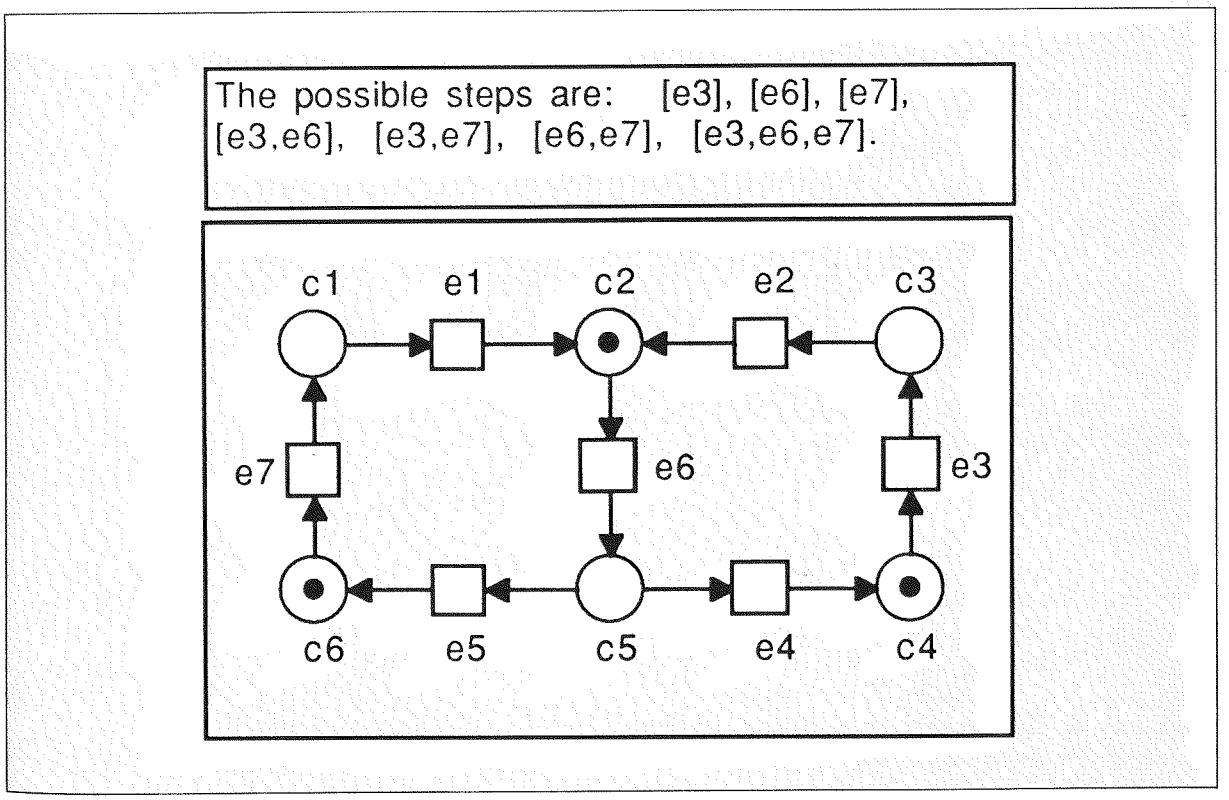
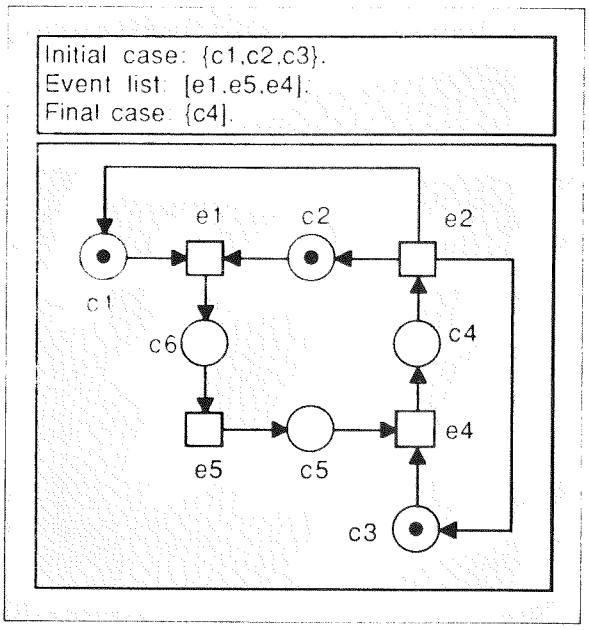


Figure 9: Show the possible steps which can occur in the current case of net



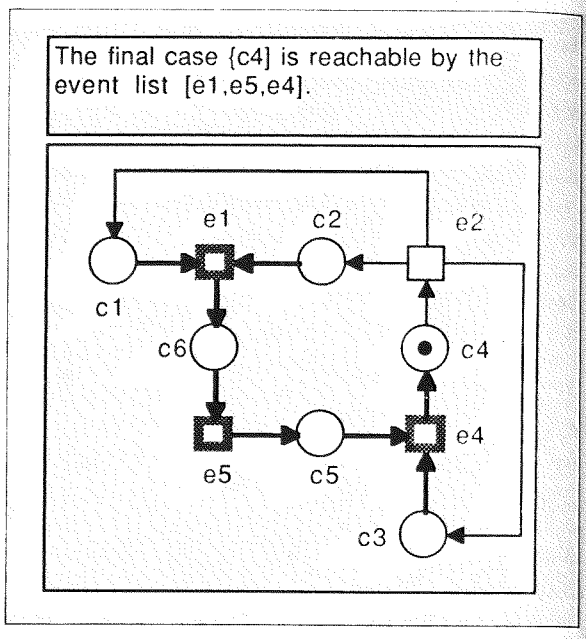


Figure 10 and Figure 11: Show the path which has to be followed to reach a particular case of the net

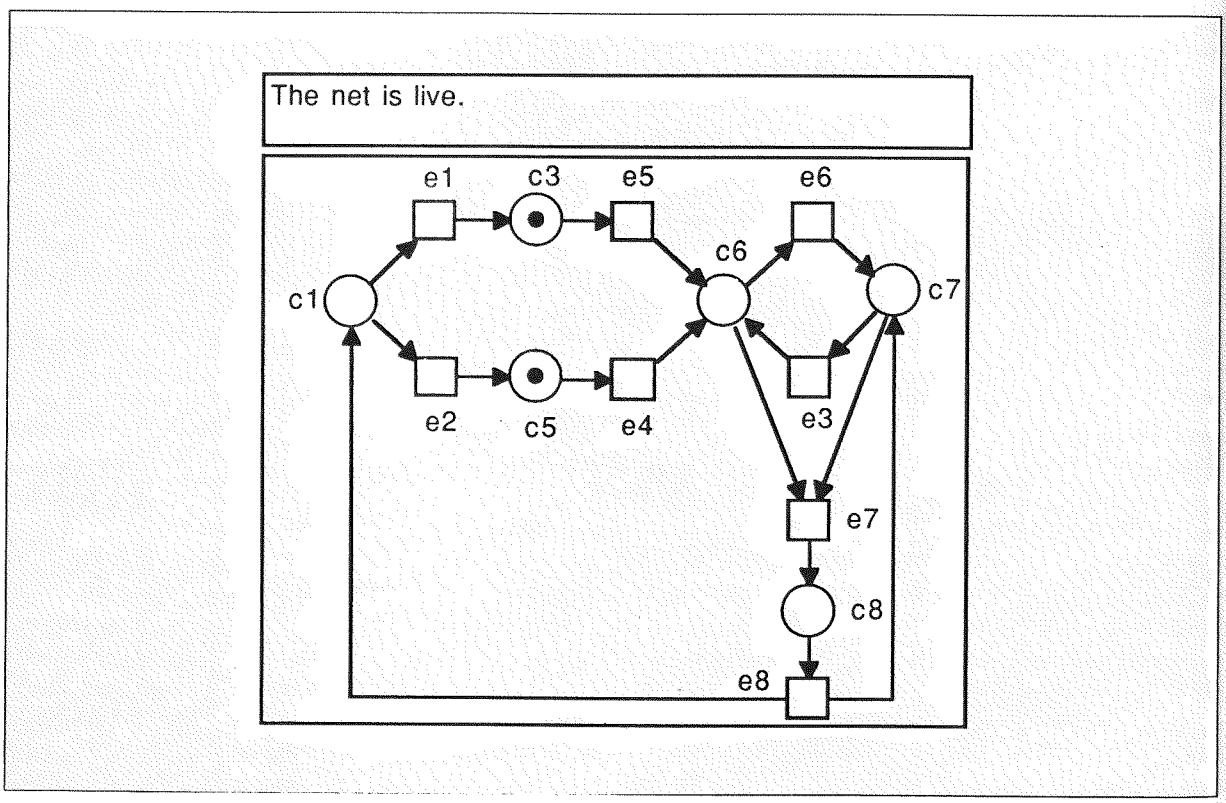


Figure 12: Shows the safety property of the net

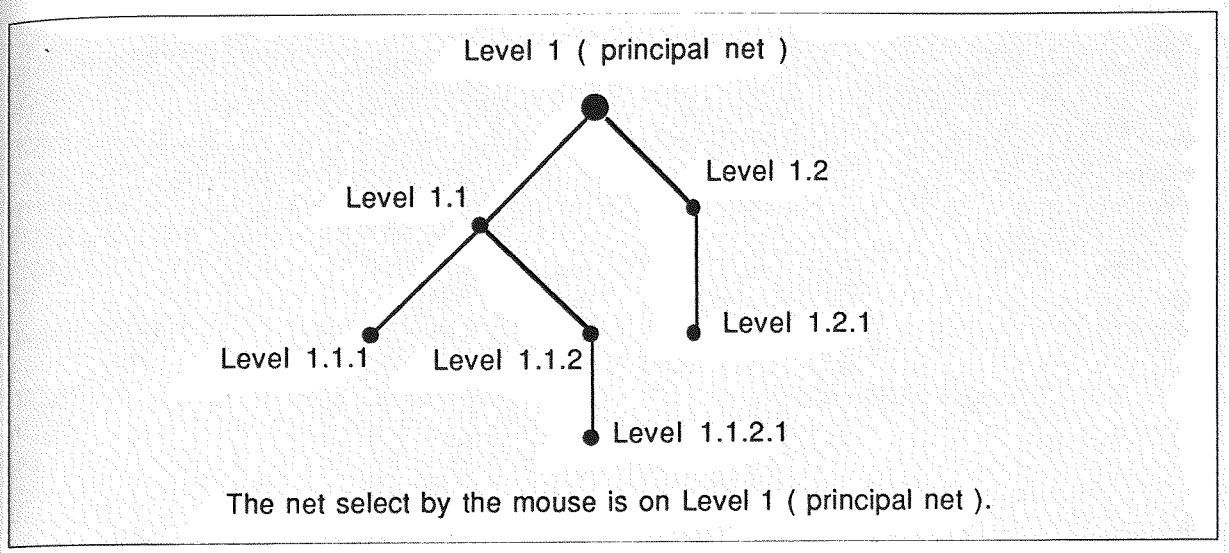


Figure 13: The hierarchical levels of a net

matica program for calculating the system invariants. Currently this interface has been implemented using the Macintosh in multifinder mode. This allows us to calculate and output the incidence matrix (or transformation matrix) using the program written with Design/OA™, switch to the Mathematica™ package and read the matrix, calculate the invariant vectors and output them to a file and then switch back to the first program and interpret the results. This method functions quite satisfactorily but requires that the user manage the file input/output and the running of the correct Mathematica™ procedures. This of course may lead to errors.
In future it is hoped that this interface can be rendered more invisible to the user, therefore making the application easier to use and less error prone. It is also possible to write these procedures for the mathematical calculation in C and include them as part of the application written under Design/OA™. Efficient algorithms for this purpose already exist, for example see [12]. We are not considering this approach currently as Mathematica™ has proved itself quite efficient to date, in this type of calculation.

4. CONCLUSIONS AND FUTURE DEVELOPMENTS

HYPERNET has been conceived and realized, with regard to the data structures utilized and the organization of the program code, in such a way to facilitate future development. The foreseeable extensions have a double aspect: in terms of graphical functionality (Net Editor and the Hypertextual Features) and in terms of the analysis of the properties of Petri Nets.
To increase the efficacy of the Net Editor and facilitate the construction of graphically complex nets we forsee the realization of a graphical feature that allows the user to create diagrams, according to a predetermined shape (star, rombus, hexagon, etc.).
To aument the power and flexibility of the Hypertextual interface we forsee the strengthening of the Net Editor so that, especially when editing the textual information associated to objects, it resembles more closely a word-processing environment.
Regarding the Net Analyser, we forsee the extension of the existing algorithms plus the addition of new ones. We expect to extend the analysis of CE-nets to P/T-nets and the extension of the PrT-net formalism used for annotating nodes and arcs thus increasing the abbreviation and modeling power of this type of net. Also, for CE-nets, we hope to implement an automatic simulation that allows the user to better visualize the dynamics of the net.
To increase the analysing power of HYPERNET we hope to generate more invariant vectors and invariants of diverse types. Once we have more invariant vectors at our disposition it would be interesting to automate their interpretation, helping the user to understand their significance with regard to the designed net and using

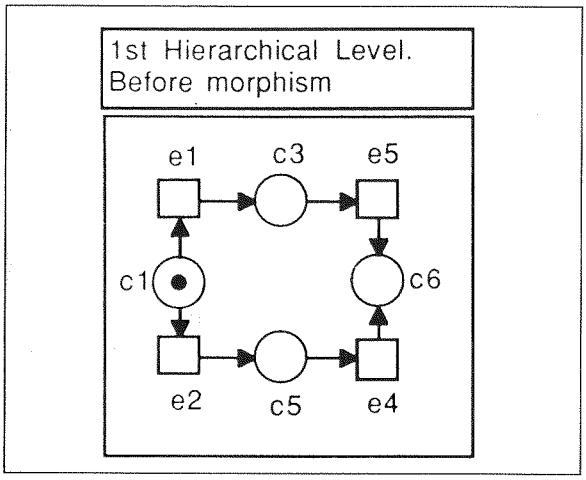


Figure 14

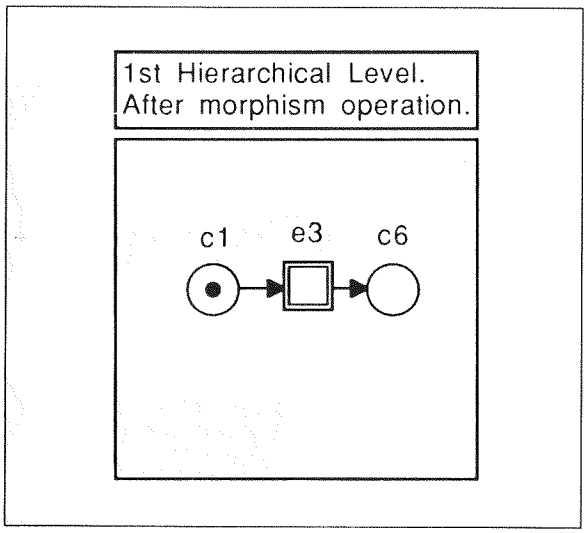


Figure 15

them to demonstrate properties of the net
Finally, a future improvement of the current program structure, that was hinted at earlier, would be to improve the current interface with Mathematica. It is hoped that with the release of version 7.0 of the Macintosh Operating System which provides for inter-process communication or with the release of a linkable version of the Mathematica™ shell that the current situation may be improved. This would help to "hide" the existing interface with Mathematica™, to some extent, thus improving the ease of use and efficiency of the application.

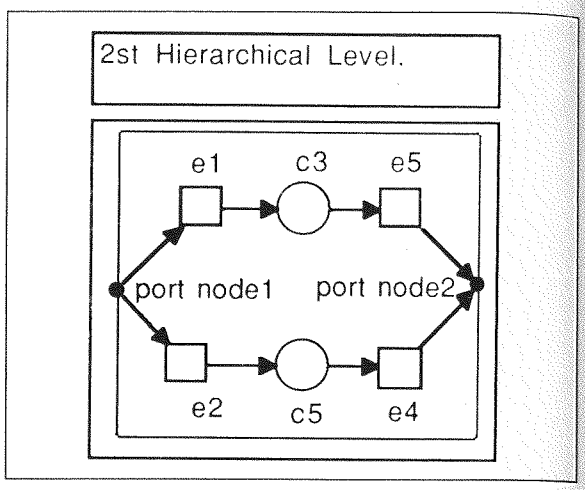


Figure 14, 15, 16: Show how morfisms are handled in Hypernet

Acknowledgements

The authors wish to thank Prof. G. Degli Antoni for his collaboration and indispensable help and encouragement during the realization of this project.
We would also like to thank, in particular, Prof. F. De Cindio and Prof. A. Pagnoni for their assistance and useful suggestions for the improvement of the current application.

5. BIBLIOGRAPHY

[1] Best, E., Fernandez, C., NOTATIONS AND TERMINOLOGY ON PETRI NET THEORY, Advanced Course on Petri Nets in September 1986

[2] Bondi, M., PROPRIETÀ DELLE RETI DI PETRI CON UNO STRUMENTO IPERTESTUALE, Conference proceedings of Verso la Communicazione Elettronica, Grande Fiera d'Aprile 1989

[3] Brauer, W. (ed.), NET THEORY AND APPLICATIONS, Lecture Notes in Computer Science Vol. 84, Springer-Verlag, 1980

[4] Norin, R. et al., DESIGN/OA - OPEN ARCHITECTURE DEVELOPER'S MANUAL, Meta Software Corporation, 1988

[5] MATHTYPE, Design Science Inc., Long Beach, CA 90803, 1987

[6] Genrich, H.J., THE ANALYSIS OF DISTRIBUTED SYSTEMS BY MEANS OF PREDICATE/TRANSITION-NETS. Lecture Notes in Computer Science Vol. 70: Semantics of Concurrent Computation, Kahn, G. Springer-Verlag, 1979

[7] Genrich, H.J and Lautenbach, K, ELEMENTS OF GENERAL NET THEORY, Lecture Notes in Computer Science Vol. 84: Net Theory and Applications, Brauer, W. (ed.) Springer- Verlag, 1980

[8] Genrich, H J and Lautenbach, K., SYSTEM MODELLING WITH HIGH-LEVEL PETRI NETS, Theoretical Computer Science, No 13, 1981

[9] Genrich, H J. and Lautenbach, K., S-INVARIANCE IN PREDICATE/TRANSITION NETS, Informatik-Fachberichte 66: Application and Theory of Petri Nets, Selected Papers from the Third European Workshop on Application and Theory of Petri Nets, Varenna, Italy, September 1982, Pagnoni, A and Rozenberg, G. (eds.). Springer-Verlag 1983

[10] Genrich, H J., PROJECTIONS OF C/E SYSTEMS, Lecture Notes in Computer Science Vol. 222, Advances in Petri Nets 1985, Rozenberg, G., Springer Verlag, 1986

[11] Genrich, H J., PREDICATE/TRANSITION NETS, Advanced Course on Petri Nets, Paper No. 4. Bad Honnef, September 1986

[12] Kujansu, R and Lindqvist, M., EFFICIENT ALGORITHMS FOR COMPUTING S-INVARIANTS FOR PREDICATE/TRANSITION NETS, Report series B, No. 29, Helsinki University of Technology, Digital Systems Laboratory, Espoo, November 1984

[13] Lautenbach, K and Pagnoni, A., INVARIANCE AND DUALITY IN PREDICATE/TRANSITION NETS AND IN COLOURED NETS, Arbeitspapiere der GMD, Nr. 132, February 1985

[14] Memmi, G and Vautherin, J. COMPUTATION OF FLOWS FOR UNARY-PREDICATES/TRANSITIONS-NETS, Proceedings 5th Eoropean Workshop on Applications and Theory of Petri Nets. Aarhus, 1984

[15] METADESIGN , Meta Software Corporation, 1989

[16] MICROSOFT WORD, Microsoft Corporation, 1989

[17] Moslemie, A., TOWARDS AUTOMATIC INTERPRETATION OF S-INVARIANTS OF PREDICATE/TRANSITION NETS, Report series A, No. 6, Helsinki University of Technology, Digital Systems Laboratory, Espoo, August 1988

[18] Petri, C.A., KOMMUNIKATION MIT AUTOMATED, Ph.D.Thesis, Insitut fur Instrumentelle Mathematik, Bonn, FRG, 1962

[19] Petri, C.A., INTRODUCTION TO GENERAL NET THEORY, Lecture Notes in Computer Science Vol. 84: Net Theory and Applications, Brauer, W., Springer-Verlag, 1980

[20] Pierotta Sergio, Pardi Alessandro, ANALISI DEL TESTO DELLA LEGGE N. 898 DEL 1970, unpublished internal report, Dipartimento di Scienze dell' Informazione, Milano 1988

[21] Reisig, W., PETRI NETS - AN INTRODUCTION, EACTS Monographs in Theoretical Computer Science, Springer-Verlag, 1985

[22] SUPERPAINT, B. Snider, Silicon Beach Software, Inc., 1986

[23] Borenstein, P. THINK'S LIGHTSPEEDC USER'S MANUAL, Symantec Corporation, 1988

[24] Wolfram, S., MATHEMATICA - A SYSTEM FOR DOING MATHEMATICS BY COMPUTER, Addison-Wesley, 1988

APPENDIX

Definitions

CE-net A triple N=(B, E; F) is called a CE Petri net iff:

(a) $(B \cap E) = \varnothing$;

(b) $F \subseteq (B \times E) \cup (E \times B)$;

(c) $(B \cup E) \neq \varnothing$;

(d) $dom(F) \cup cod(F) = (B \cup E)$.

F F is the flow relation di N.

Arc The elements of F are called arcs.

Conditions The elements of B are called Conditions.

Events The elements of E are called Events.

dom(F) $dom(F) = Z = \{ x \mid \exists\ y : (x, y) \in F \} = F \cap (B \times E)$.

cod(F) $cod(F) = Q = \{ x \mid \exists\ y : (y, x) \in F \} = F \cap (E \times B)$.

X The set $X = (B \cup E)$ is the set of elements of the net.

P The set $P = (Z \cup Q)$ is the adjacency relation.

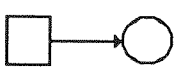
Grafical representation of a net

Condition are represented by circles

Events are represented by squares

The flow relation F is represented by arrows between the elements of X, in such a way that:

$(x, y) \in (F \cap (B \times E))$

$(x, y) \in (F \cap (E \times B))$ 

Definitions

Let N=(B,E;F) a net and $x \in X$. Then

Pre-set of x $\{ {}^\circ x = \{ y \mid (y, x) \in F \}$.

Post-set of x $\{ x^\circ = \{ y \mid (x, y) \in F \}$.

S-simple net N is S-simple iff $\forall\ c_1, c_2 \in B$:

$(({}^\circ c_1 = {}^\circ c_2) \wedge (c_1{}^\circ = c_2{}^\circ)) \Rightarrow (c_1 = c_2)$.

T-simple net N is T-simple iff $\forall\ e_1, e_2 \in B$:

$(({}^\circ e_1 = {}^\circ e_2) \wedge (e_1{}^\circ = e_2{}^\circ)) \Rightarrow (e_1 = e_2)$.

Simple net N is called simple iff is both S-simple and T-simple.

Indep Let $e_1, e_2 \in E$. Then e_1 indep e_2 iff

$(({}^\circ e_1 \cup e_1{}^\circ) \cap ({}^\circ e_2 \cup e_2{}^\circ)) = \varnothing$.

Side Condition If $e \in E$ and $c \in ({}^\circ e \cap e^\circ)$ then c is called a side condition of e.

Pure net A net N is called a pure net iff

$\forall\ (c, e) \in (B \times E)$: $(c, e) \in F \Rightarrow (e, c) \notin F$.

Case $k \subseteq B$ is called a case.

K The set of cases is denoted by K= *P* (B), where *P* (B) is the powerset of B.

Concession Let $e \in E$ and $k \in K$. Then e has concession in k

(or e is enabled at k) iff ${}^\circ e \subseteq k$ and $e^\circ \cap k = \varnothing$.

Follower case Let $e \in E$ and $k \in K$. If e is enabled at k then

$k' = (k \setminus {}^\circ e) \cup e^\circ$ is called the follower case of k under e.
We also say that k' results from the occurance of e in the case k and we write k[e>k'.

Step Let $G \subseteq E$ and $k, k' \in K$. We say that k' is reached from k by the step G and we write k[G>k' iff:

(i) $\forall\ e_1, e_2 \in G$: $(e_1 \neq e_2) \Rightarrow (e_1$ indep $e_2)$.

(ii) $\forall\ e \in G$: e is enabled at k and $k' = (k \setminus {}^\circ G) \cup G^\circ$.

Reachability Let $(k,k') \in r$. r is a relation $\Leftrightarrow \exists\ G \subseteq E : k[G>k'$.

Then $R = (r \cup r^{-1})^*$ is called full reachability relation.

Case class C, called the case class of the CE-net, is an equivalence class of the reachability relation R such that:

(i) $\forall\ k_1, k_2 \in K$, if $k_1 \in C$ and $(k_1, k_2) \in r$ then $k_2 \in C$.

(ii) $\forall\ k_1, k_3 \in K$, if $k_3 \in C$ and $(k_1, k_3) \in r$ then $k_1 \in C$.

(iii) If $k \in K\text{-}C$ then $\forall\ c \in C\ (k, c) \notin r \wedge (c, k) \notin r$.

(iv) $\forall\ c_1, c_2 \in C,\ (c_1 = c_2) \vee \exists\ n>1$

such that $(c_1, c_2) \in (r \cup r^{-1})^n$

Contact Let $e \in E$ and $c \in C$, then e is in contact iff:

$^\circ e \subseteq c \wedge (e^\circ \cap c) \neq \varnothing$

Conflict Let $e_1, e_2 \in E$ and $c \in C$, then e_1 and e_2 are in conflict at c iff $c[e_1 >$ and $c[e_2 >$ but not $c[\{e_1, e_1\} >$.

Cyclic A net N is called cyclic iff $\forall\ c_1, c_2 \in C : c_1\ (r^*)\ c_2$.

Live A net N is called live iff $\forall\ c \in C$ and $\forall\ e \in E$, $\exists\ c' \in C : c\ (r^*)\ c'$ and e is enabled in c'.

Safe A net N is called safe iff $\forall\ c \in C$ and $\forall\ e \in E$:

$(^\circ e \subseteq c \Rightarrow e^\circ \subseteq B\text{-}c) \wedge (e^\circ \subseteq c \Rightarrow {}^\circ e \subseteq B\text{-}c)$.

Definitions

Morphism Let $P_i = (F_i \cup F_i^{-1}) \cap (S_i \times T_i)$ a adjacency relation and $f: X_1 \to X_2$ a function. The tern (N_1, N_2, f) is called morphism iff f observes P and F, that is preverse $(P \cup id)$ and $(Q \cup id)$:

(i) $f(P_1) \subseteq (P_2 \cup id_{X_2})$.

(ii) $f(F_1) \subseteq (F_2 \cup id_{X_2})$.

is a solution of linear system of equations ($C^T \bullet i) = \Delta$ is called a name S- invariant.

(Name) T-Invariant If C is the incidence matrix of a Pr/T net then every vector j which is asolution of linear system of equations ($C \bullet j) = \Delta$ is called a name T-invariant.

Definitions

Pr/T net Let **L** be a first-order language. Then the class PRT_L consists of marked annotated nets MN = (N, ¬, A, M_0), where:

1. N = <S,T;F> is the underlying directed net, where:
   (a) $S \neq \emptyset$ is the (finite) set of places,
   (b) $T \neq \emptyset$ is the (finite) set of transitions,
   (c) $T \cup S = \emptyset$
   (d) $F \subseteq (S \times T) \cup (T \times S)$ is the flow relation,
2. ¬ is a structure for L (i.e. a support for N)
3. A = <A_S, A_T, A_F> is the annotation of N, where:
   (a) A_S is an injective mapping of S into the set of variable predicates

   Y belonging to **L**
   (b) A_T is a mapping of T into the set of formulae of **L** constructed from P and W only,
   (c) A_F is a mapping of F into the set of symbolic sums of tuples of **LX** such that for an arc (x,y) ∈ F leading into or out of a place s and n being the index of the predicate annotating s, $A_F(x,y)$ is in $\mathbf{LX^n}$.
   4. M_0 is a mapping from S into the set of symbolic sums of tuples of constants, **LU**, such that if n is the index of the predicate annotating *s*, then $M_0(s) \in \mathbf{LU^n}$.

Incidence Matrix Let MN = $< N, \neg, A, M_0 >$ be a Pr/T net. The incidence matrix of MN is a matrix C such that for all $s \in S$ and $t \in T$, C(s,t) is defined as follows:

$$C(s,t) = \begin{cases} -A_F(s,t) & \text{if } (s,t) \in F \setminus F^{-1} \\ A_F(s,t) & \text{if } (s,t) \in F^{-1} \setminus F \\ A_F(t,s) - A_F(s,t) & \text{if } (s,t) \in F \cap F^{-1} \\ 0 & \text{otherwise} \end{cases}$$

Self loops are not possible.

Tranformation Matrix Let C be a incidence matrix. Then the total projection of C, denoted with $|C|$, is the transformation matrix of a PrT net.

(Name) S-Invariant If C is the incidence matrix of a Pr/T net then every vector i which is a solution of linear system of equations $(C^T \bullet i) = \Delta$ is called a name S- invariant.

(Name) T-Invariant If C is the incidence matrix of a Pr/T net then every vector j which is asolution of linear system of equations $(C \bullet j) = \Delta$ is called a name T-invariant.

A HYPERMEDIA INFORMATION SYSTEM TO MANAGE THE ACTIVITIES IN A RESEARCH INSTITUTE

Mario Allegra, Onofrio Di Giuseppe, Silvana Mangiaracina
Istituto di Tecnologie Didattiche e Formative - CNR
Palermo

SOMMARIO

In questo articolo presentiamo un sistema ipermediale per la gestione delle attività di un istituto di ricerca. Il sistema si basa sulla metafora della realtà artificiale, per cui anche un utente che non abbia alcuna competenza d'informatica, può visitare l'interno dell'istituto e conoscere le sue attività e la sua storia, navigando attraverso la libreria, l'organigramma, il bilancio, i progetti scientifici, ecc. Descriveremo prima la struttura delle attività, mettendo in evidenza gli aspetti che si prestano ad essere automatizzati, sia della vita amministrativa che di quella scientifica. In secondo luogo si vedrà come è stato possibile tradurre la struttura formale in un sistema ipermediale e si discuteranno le soluzioni adottate, l'interfaccia utente e le "utilities" ipertestuali che consentono l'aggiornamento ed il mantenimento del sistema senza alterarne la struttura. Infine si metteranno in luce le prospettive e le possibilità di utilizzo future.

ABSTRACT

Dynamic hypermedia up to today have been used in cooperative writing, collaboration work, on-line publishing and authoring courseware. In this paper we present an example of information system for managing and supporting activities in a research institute. The kind of information used in an institute is of different types; furthermore, some data have to be updated frequently. The system helps the workers to fill, to update and to have access to information. We believe hypermedia is suitable in managing these data effectively and in this paper we'll examine how it can be used in this kind of organization, instead of stating hypermedia technology per se. We'll then describe the structure of HyperITDF system and its use in the institute.

1. INTRODUCTION

As part of an on-going effort to develop computer application appropriate for automation in a recently established institute of the National Research Council (CNR) and to make its aspects well known to the new staff, the Istituto di Tecnologie Didattiche e Formative (ITDF) has developed a prototype hypermedia information system to manage a research institute of CNR.
The hypermedia system is based on artificial reality metaphor. With the increase in computer quality the interaction between the computer and its users increasingly resembles an interaction of the users themselves with an artificial reality. The users take actions, send messages, modify things, and receive commands and messages exactly as in the ordinary world. Moreover, with the increase of graphical capability, the computer world is frequently very similar to the real one. The artificial reality metaphor offers a methodology of interaction between the communication system of man and that of the computer [1], and [2]. In a sense, the user, even without any ability in computer science, enters the reality projected on the computer screen and navigates through it, or directly modifies it. In accordance with this model, the user of our hypermedia system can visit the interior of the institute [3], and get to know its

activities and its history, navigating through the library, the staff of the institute, the budget management, the present and past scientific projects,....
In this paper we will first describe the frame of the activities in a research institute, underlining those activities which are suitable for automation, both in the scientific and in the administrative field. Next, we will describe how it is possible to translate the formal frame to a hypermedia system, and we will discuss the introduced solutions, the user interface and hypertext utilities which allow updating and maintenance of the system without altering its structure. Finally, we will focus on perspectives and future development and use.

2. THE FRAME OF THE ACTIVITIES

The range of activities in a CNR Institute is wide and various; we can roughly distinguish five areas of activity: the management, the administration, the research, the secretarial, the scientific advisory committee. We present in figure n.1 a scheme of the main activities of each area. These activities are related one to the other by different kind of links, and we will see how they are suitable for automation later on. During some of these activities, it is necessary to make previsions (in a variable range of time) and, at the end or during the established time, to carry out the revision of the work done. These activities, such as budget estimate and final balance, which include forecasts and revisions are highlighted in the figure with a double rectangle.
As illustrated in the figure, the management is responsible for coordinating the activity of the institute, for supervising the scientific research, for planning human resources and funds. Besides that, the management is responsible for the institute staff, divided into research staff (technicians, researchers, grantees, external collaborators), administration and secretarial staff.
Research activity is carried out by groups which work on one or more projects; the group leader has to evaluate the resources needed and present a description of the activity for each project, containing information about new publications, new technologies, collaborations with other centers or universities, estimated or obtained funds .
The administration deals with personnel management (that is, it calculates personnel attendances, manages travelling expenses, fills in the personnel data base...), the management of general expenses, that is expenses of common interest, not assigned to a single project, the elaboration of an inventory of the institute property, maintaining relationships with suppliers.
The Scientific Advisory Committee, made up of the management, and several internal and external members, is the organ that supervises the scientific activity of the institute : it has to promote new research trends and approve the research relations supplied by the management and the group leaders, discuss the balance of the institute and award temporary contracts. The committee meets at least twice a year, and its activity is reported and sent to the central board of managements of the CNR. Members of the committee are nominated every four years.
The secretariat supports the management, and deals with the incoming and outgoing mail. Mail is filed using inward-outward mail registers, where every letter is given a progressive number. The arrival/departure date, the type, the subject, the sender/addressee is recorded.

3. HYPERMEDIA SYNTHESIS THROUGH FRAME IMPLEMENTATION

In this section of the paper we'll show how the described frame has been translated into the informative system HyperITDF, using Hypercard™ [9].
Still with reference to fig. n.1, in symmetry with the areas of activity, there are in the system the same number of access points, which allow users to carry out their assigned activities.
A first cluster of homogeneous information is obtained in the system using the stacks of Hypercard in symmetry with each activity. Another correlation among the information is obtained through several kinds of links which allow both the navigation into the system and the flow of information. The first ones are called "cold links", while the second ones are "hot links". Furthermore, using the terminology introduced by Catlin [4], there are "warm links", which allow a user not only to traverse a link from a selection in one document to a selection in another document but also to transport data across the same link.
The user can navigate through the system across the information regarding his duties; can update them; fill new data, or require their elaboration. Such operations leave the hypertext structure of the system unchanged.

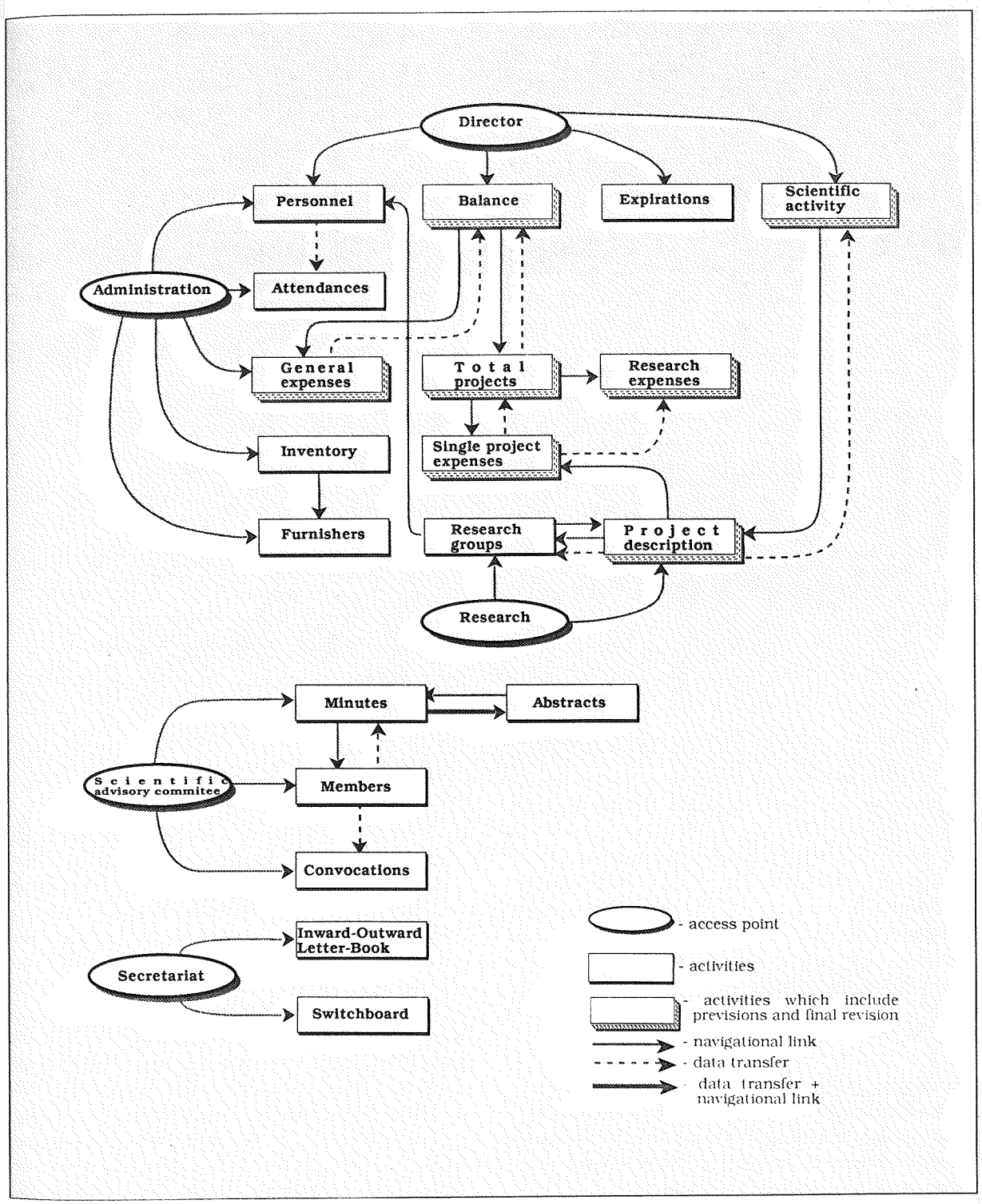


Figure 1: The frame of the activities

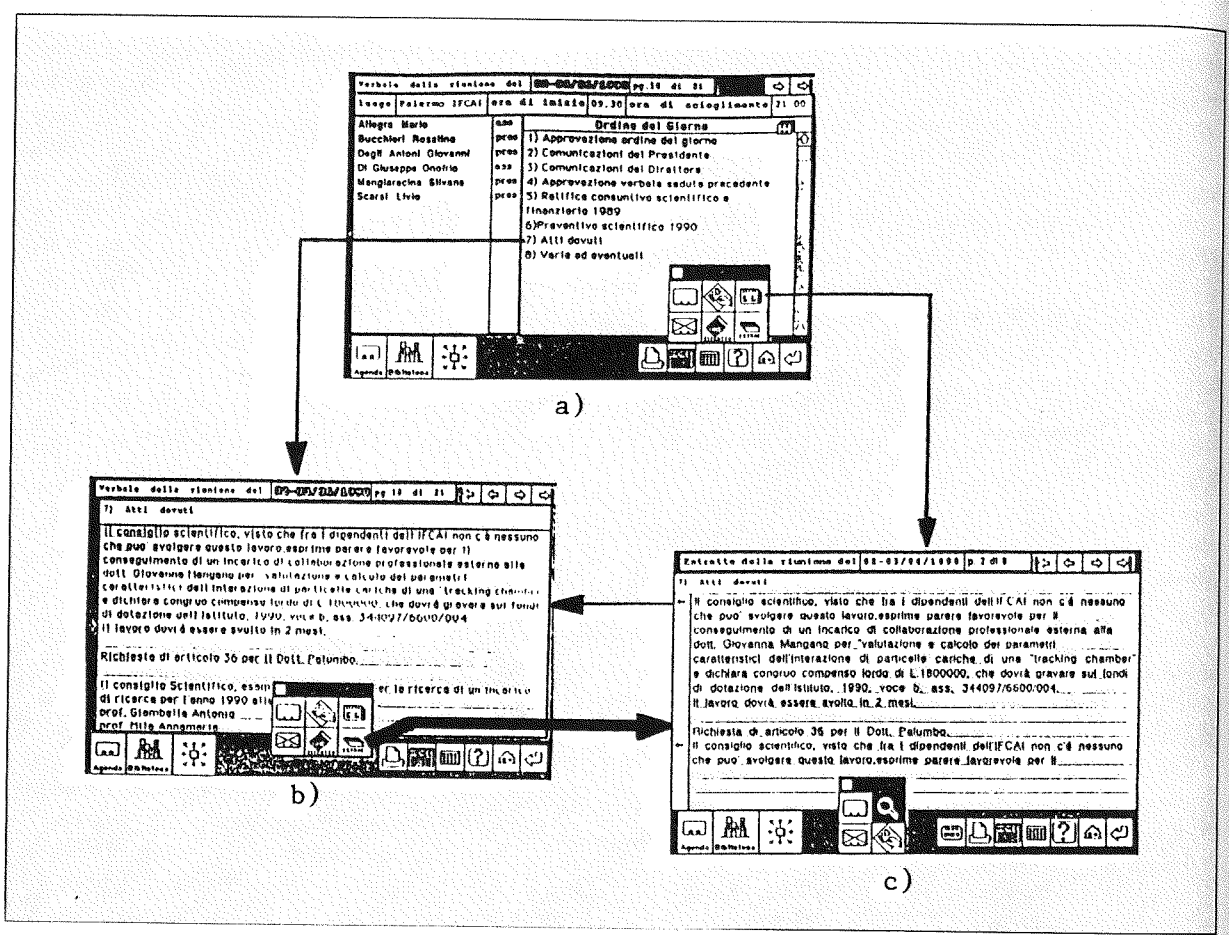


Figure 2: Links between scientific council and abstracts

In fact, when new data are added, new context dependent links are made, in an automatic and entirely transparent way to the user.
Besides, in HyperITDF there is information which guides you to the knowledge of the institute, what it contains and how it is physically structured. Using the metaphor of navigation through the layout of the institute, there is a representation of the building, of the rooms, of the objects and people that are in each room.

3.1 The user interface

In order to make the user interface intuitively simpler, we have chosen a homogeneous graphic representation in all parts of the system. Just as books have a unity of interface because everything is printed on the same paper, in the same way hypertext should preserve such a unity avoiding a needless multiplication of data types [7].
In HyperITDF the user always has available on the screen the same kind of information, that helps him to solve the location problem (we adopted some conventions of print paper as titles, subtitles and pagination). Also available is the same set of iconic buttons, divided into two groups of action: in the first group there are navigation buttons, accessible from anywhere in the system, as address-book, library, map of the institute, browsing and backtracking buttons, while the second group provides utility buttons, as print of reports, online help, a command window. The command window is used by the user who not only browses, but may also update information; it contains a set of differentiated buttons (not all possible actions are allowed in every part of the system) which allow a new page to be added,

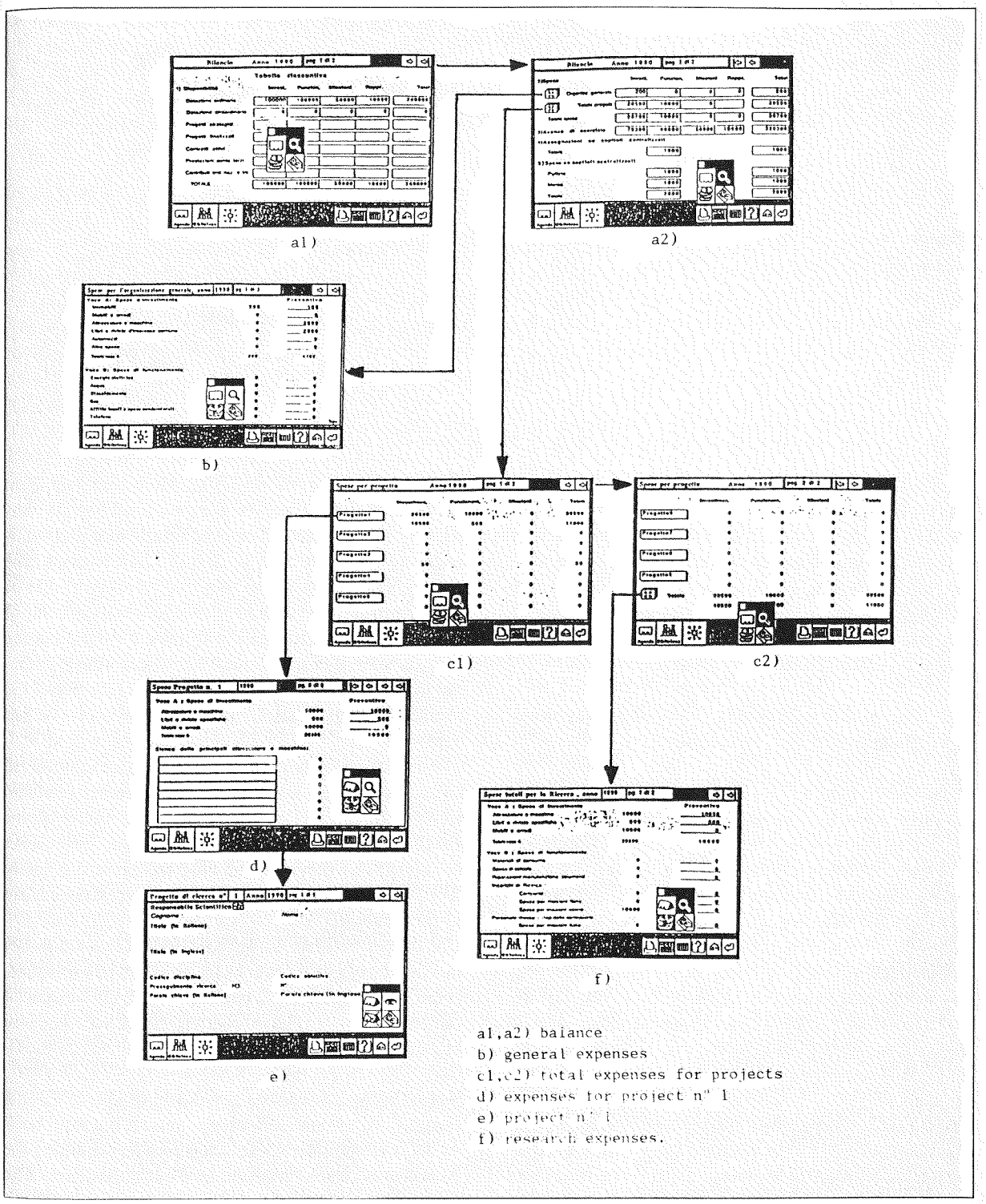


Figure 3: An example of the budget

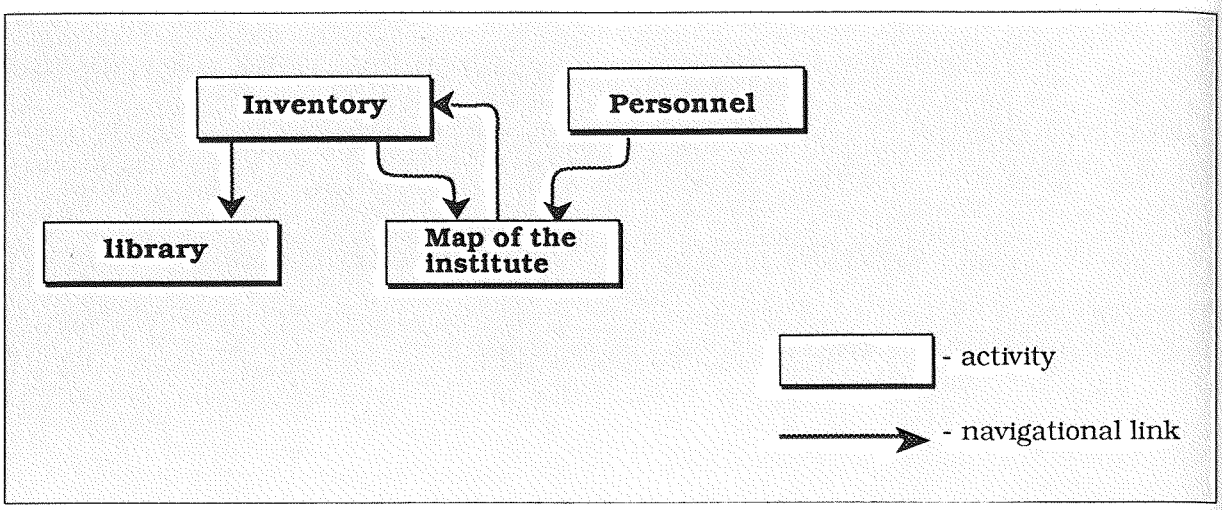


Figure 4: Links between the inventory, the map of the institute and the personnel

to cancel it, to switch from read to write mode, to add or cancel annotations.
Some hypertext utilities are also available to the user, to allow him to attach annotations over the text or data, or to create, activating warm links, abstracts from a text linked to the text itself. This is achieved by means of buttons (to which scripts and text fields are automatically associated) positioned where the user desires.

3.2 The Scientific Advisory Committee: the members, the minutes, the convocations

HyperITDF contains a file of the members of the scientific advisory committee and a minutes file.
The file of the members: it contains recorded information about members, i.e. their position inside the S.A.C., and regards both members in charge at the moment and members that have been in charge in previous years. It is possible to navigate both through the past members and through the present members.
The minutes management: in HyperITDF we developed a tool to edit and print a new minute according to a determined outline and help to create abstracts, containing only the most significant actions of the committee meeting.
The user always enters the most recent minute and can choose, selecting an item from a list of meetings, the minute or the abstract he wishes to read. A new minute and its related abstract are created using the buttons of the command window. After a last review of the minutes it is in fact possible to extract the most salient passages and generate the summary. Using the "extract..." button, selected parts of text will be automatically copied in the abstract stack. We introduced the button "end of extract..." which is used after one or more "extract..." operations, to allow the creation of paragraphs, containing pieces of selected text about the same subject. It brings about the creation of a navigational link between the paragraph just extracted and the initial page of the minute, from where the extraction had started.
The user can browse in the stack of abstracts, generated and augmented by further extractions, and he can obtain an immediate view of all the decisions taken by the advisory committee during any meeting, and, whenever he wants to know the reasons, or, generally, more detailed information, to straightly look up in the minute of the meeting.
One of the problems in dynamic hypertexts is the fact that a piece of text, with a source or target link attached, could be updated, cancelled or moved. In the creation of our abstract stack, this problem does not arise because links are created from static text that cannot be modified.
HyperITDF also supplies a tool to help to convoke the meetings of the advisory committee : it is a note-book where it is possible to store information regarding the days in which each member can attend the meeting. The list of members to be convoked is automatically taken from the file of present members. This tool is not a file

of meetings, in fact, when a new meeting is required, the old one is cancelled.

3.3 The management and the research: the balance, the expirations, the scientific activity, the research groups, the projects

The balance sheet - The management of the balance sheet is made up of 5 stacks: Balance, General Expenses, Total Expenses for the Projects, Research Expenses, Project Expenses.
Inside every stack, the cards are put in order by year; the user directly reaches the current year, but he can easily look up the previous years' information. The stacks are connected to each other by means of cold and hot links.

The links between the various stacks directly connect information concerning the same year.
The Balance contains resumptive information about the available funds of the institute, about the totals of the general expenses, about the totals of the expenses for research and the eventual cash in hand.
General Expenses and Project Expenses contain instead more detailed information about the expenses, divided into 4 classes: working, property for inventory, missions and representation.

As you can see in fig. n.1., while the navigation gives you the possibility to examine the balance from general to detail, the information flow goes from detail to general; when you update or add information into the stack General Expenses or Expenses for Project, all data depending from it are updated automatically, not only into the present stack but also into the others.
In fig. n.3 you can see an example of navigation through the balance in HyperITDF. The management directly reaches the resumptive information about the balance, but he can however arrive at the details of the expenses through one or two deepening levels. In fact through a direct link he can reach the details of the expenses, while for the details of the Project Expenses there is an intermediate step, given by the cards for each total of the expenses for the projects.

In the budgeting phase the system gives the person responsible for the projects and the administrator the possibility to fill the estimate of general expenditures and the estimate of necessary expenditures for each project and can immediately verify the whole effect.
When real expenses are filled, the users will be able to reach the corresponding estimate, verifying in this way, the cash in hand, at any time.
The expirations: to have a general view of the expirations to observe, the management can use the stack "expirations". Each project, or object, has a final expiration which is made up of several intermediate steps, that have exact expirations too.
Thus "expirations" is an "ad hoc" annotation system for the management. On these annotations, divided by object, there are a couple of functions of "intelligent note-book": every time a page, concerning one object is opened, the nearest (compared to today) expiration is pointed out.

Another function allows a resumptive page (the first) to be reached, where the list of all objects present in the stack appears, and to each object is associated the expiration date of the closest sub-object. Besides, in the first page you can select by means of the mouse one of the objects of the list, and you can go, by means of a dynamic link, to the page concerning that object.
Besides it is possible to add notes to the text of every page [6]. Which means that you can annotate in a hidden field, explicative notes relating to the text, or something that you have to remember relating to a particular action. The note can be inserted in an exact point of the page, deleted, restored visible or hidden by means of an iconographic button that indicates its presence. The two functions of filling and cancelling of a note are recalled by the window control.
The scientific activity, the groups and research projects: Like in the balance management, the scientific activity has two deepening levels.
The first level gives an over-all vision of the institute activity and helps the management to present programs and reports on the work done. The system gives information about research trends, eventual future joint-projects, organization of training activities, such as schools, courses, seminars, organization of conferences, publishing of papers, technical reports and internal memories, collaborations in scientific magazines. A general table of the tasks of all the scientific personnel is present; this table is obtained from the single project data, using hot links.
The second level examines carefully the groups and project activities and the planning of the tasks over any

single project.

3.4 The administration and the secretariat: the inventory and the suppliers, the library, the personnel and the inward/outward letter-book

The attendances: The system provides a tool for the calculation of the attendances and the hours of work that the employees effected, and keeps the comprehensive situation of the holidays, of the days of illness, etc, automatically updated in a period of one year.
The inventory, the suppliers and the library: They are three archives that manage data about property and about the firms that furnish materials or that give services to the institute and about the books of the library. The links between the inventory and the map, connect information about furniture or machines directly with the place where they are (fig. n.4).
The personnel: The personnel of a CNR institute is divided into 5 categories: research, administration, secretariat personnel, grantees and external collaborators. All the information, such as private data, matriculation number, title, position ... are stored in the stack of the personnel. It is possible to navigate through the curriculum, the training or the interest area of a person or, through the institute map, to see where that person physically is inside the institute.
The inward/outward letter-book: the inward/outward letter-book in HyperITDF system has the same functionality of the paper one. It is divided into two registers. The pages of the letter-book are graphically like the ones in the paper letter-book and it is not possible to tear them; when a new page is required the system automatically assigns the next new numbers. Classification by correspondence types is available on request.
It is possible to create, and then to go through, a correspondence path of inward/outward letters related to the same subject. Each letter is associated to three reference numbers; two numbers are the references to a previous or a following letter, and the other one is the reference to an answer letter. In the first case, for example, when the user writes the previous reference number of a letter in the book, the system automatically updates the reference of the "following" number in that previous letter. Selecting these numbers the user can navigate through the next (or previous) letter and back. This is an example of bidirectional dynamically created link.

3.5 The map of the institute

We are integrating in HyperITDF a graphical and interactive description of the building and the objects contained inside the institute. Such a description has been developed by ARS Informatica snc, Milano, and we started a collaboration in order to link, in a bidirectional way, the objects of inventory and the personnel to their physical location inside the institute (fig. n.4). "Walking" through the rooms of the institute it will be possible to get information about the objects and the people working there, and vice versa.

4. USE OF THE SYSTEM AND FUTURE DEVELOPMENT

HyperITDF is a prototype now in use in the *Istituto di Tecnologie Didattiche e Formative*. It is useful in the carrying out of some activities and it has a good performance; since it has been used for a short period of time, many data haven't yet been entered. This shouldn't be a problem for retrieval performance, because the search functions used in the archives of the system have been implemented by means of the find command that guarantees good performance also in presence of a lot of megabytes of information. In order to give the user the possibility of more articulated queries (in particular for the library and the inventory) we plan on integrating the system with a data base management system.
It will be necessary to analyze the problem of periodically saving information in mass storage, while preserving navigation through the system. Optical memory support and the use of a scanner will become essential to eventually introduce a archive of all the documents of the institute: in this way, by means of the inward/outward letter-book, the user can directly have access to documents.

5. CONCLUSIONS

HyperITDF is an example of hypermedia office information system. It allows for integrating and linking in the navigational space distributed or connected activities and, according to the model of artificial reality, it provides a natural and anthropomorphic interaction with the user. Dynamic updating and maintenance of the system leave unchanged its hypertext structure.

Acknowledgements

The HyperITDF information system is an idea of prof. Giovanni Degli Antoni. He gave precious suggestions and comments on our project. We would like to thank also dr. Rosolino Buccheri of the IFCAI CNR Institute for helping us to master the knowledge of a CNR institute and Roberta Macchi of the Hypertext User Group for her collaboration.

6. REFERENCES

[1] Giovanni Degli Antoni, FROM REALITY TO HYPERMEDIA THROUGH ARTIFICIAL REALITY, Educational Media International, vol.26, n.3, Sept.1989, pp.149-152

[2] Giovanni Degli Antoni, IL COMPUTER, IL REALE, L'ARTIFICIALE, Note di Software n.40, 1988

[3] P. Valcher, M. Locatelli, D. Bianchi, R. Morandi, UN SISTEMA INFORMATIVO COGNITIVO PER IL DIPARTIMENTO DI SCIENZE DELL'INFORMAZIONE, in Prime applicazioni di Realtà Artificiale (nota interna dello Hypertext User Group)

[4] T. Catlin, P. Bush, N. Yankelovich, INTERNOTE: EXTENDING A HYPERMEDIA FRAMEWORK TO SUPPORT ANNOTATIVE COLLABORATION, Proc. of Hypertext'89, november 5-8, Pittsburgh, pp.365-378

[5] E. Yoder, USING HYPERTEXT IN A LOW FIRM, Proc. of Hypertext'89, november 5-8, Pittsburgh, pp.159-168

[6] L. De Young, HYPERTEXT CHALLENGES IN THE AUDITING DOMAIN, Proc. of Hypertext'89, november 5-8, Pittsburgh, pp.169-180

[7] T. Oren, THE ARCHITECTURE OF STATIC HYPERTEXTS, Proc. of Hypertext'87, november 13-15, Chapel Hill, pp.291-306

[8] P. Kahn, N. Meyrowitz, GUIDE, HYPERCARD AND INTERMEDIA: A COMPARISON OF HYPERTEXT/ HYPERMEDIA SYSTEMS, IRIS Technical Report Number 88-7

[9] Apple Computer inc, HYPERCARD REFERENCE MANUAL,1988

"THE HEART: ITS FUNCTIONS AND DISFUNCTIONS": LESSONS LEARNED AND OPEN QUESTIONS FROM A CD-ROM PROJECT

Alessandro Campana, Riccardo Cattaneo, Patrizia Ghislandi
Centro Televisivo Universitario
Università degli Studi
Milano

RIASSUNTO

Multimedia, ipermedia e interfacce a manipolazione diretta che usano metafore della vita quotidiana stanno modificando radicalmente il modo con cui comunichiamo e trasmettiamo conoscenza.
Ancora molto é da fare per definire una grammatica ed una sintassi del linguaggio che le nuove tecnologie e metodologie inducono ad utilizzare.
Le ricerche possibili riguardano i campi più disparati: strumenti per reperire nel modo più efficiente e rapido l'informazione desiderata; modalità per ritrovare la via nell'universo dell'informazione reso disponibile dagli ipermedia; modelli nuovi di navigazione nella conoscenza; metodi per il trattamento dell'informazione multimediale in ambiente digitale; metodologie di progetto e produzione di ipermedia.
Il presente lavoro descrive il CD-ROM ipermediale: "The heart: its functions and disfunctions", la cui realizzazione ha portato ad alcuni risultati innovativi nelle ultime tre aree di ricerca elencate, ed ha consentito di verificare in concreto idee proposte da altri autori.

ABSTRACT

Multimedia, hypermedia and direct manipulation interfaces which use metaphors taken from everyday life are radically changing the way in which we communicate.
A great deal remains to be done before a grammar of the language which the new technologies and methodologies are leading us to use can be defined.
Possible researches cover widely varying fields: tools to find the information required in the most efficient and rapid way; methods of "finding the way" in the universe of the information made available by hypermedia; new models of navigating through knowledge; methods for the treatment of multimedia information in a digital environment and methodologies of design and production of hypermedia.
This paper describes the hypermedia CD-ROM (Compact Disc read Only Memory): "The heart: its functions and disfunctions", which has led to a number of innovatory results in the last three areas of research listed above and has allowed the concrete application of ideas suggested by other authors.

1. INTRODUCTION

Multimedia means the possibility for the personal computer to communicate using, in addition to the software, different information such as text, images (natural still images or moving images, graphics or animated sequences) and audio.
"Hypermedia" means a multimedia with hypertext navigation paradigm.
Multimedia or hypermedia programs can be produced using optical memories, thanks to the high capacity they make available. Up to the present time, the most sophisticated projects have been produced in an analogical environment, that is on videodiscs.
This technology has however highlighted a number of shortcomings, such as the poor resolution of the images, linked to the standards established for television; the

lack of a standard for the delivery stations and the fact that the information recorded cannot be manipulated.
The digital alternative, which we have explored, overcomes these shortcomings and integrates all the different types of information in a single environment.
Consequently this is surely the forthcoming solution for the treatment of information and transmission of knowledge.
The digital solution entails, however, a number of problems, for example a significant part of the mass memory is occupied by the multimedia information and certain difficulties are encountered in reproducing moving sequences on the whole screen and processing the audio synchronized with the images.
These problems, together with a proposal for a new model of navigation and an interface for hypermedia, will be discussed below.

2. THE HYPERMEDIA CD-ROM: "THE HEART: ITS FUNCTIONS AND DISFUNCTIONS"

The project was carried out by the CTU (University of Milan Television Centre) in conjunction with the Department of Pathology ,University of Milan, for a COMETT 1 (Community Programme in Education and Training for Technology) programme.
The contents are:

- ANATOMY: exposure of the heart, atria, ventricles, valves (mitral, tricuspid, aortic), conduction system, sinoatrial (SA) node, atrioventricular (AV) node, AV junction, HIS bundle (HB) , bifurcating HB
- PATHOLOGY: mitral stenosys, stenosys of aortic valve

The final CD-ROM should also include the topics of embryology and physiology of the heart.
The material on the topics listed above was taken from two publications:
- Cardiac conduction blocks and pacemaking. An anatomoclinical color atlas.
By: Lino Rossi, Luigi Matturri, Antonio Lotto
Published by CLAS International
- The CIBA collection of medical illustrations.
A compilation of paintings on the normal and pathologic anatomy and physiology, embryology and diseases of the heart.
Prepared by Frank H. Netter. Edited by Frederick F. Yonkman
Commissioned and published by CIBA.
The final programme has been designed for users who already have a basic theoretic knowledge of the subjects and who wish to gain more specialized knowledge. The ideal users are students in the early years of a degree in medicine and surgery as well as training courses for paramedical staff.

The requirement of the project was an integrated management of objects of a multimedia type, in various combinations. Let us first consider a description of the different types of information under examination.

2.1 Images

There are two categories of images: highly detailed colour anatomical plates and reproduction of hystological slides obtained using a microscope. All the images are present in colour and in black and white. This has been done in order to improve the performance of the product in the consulting phase. Given the average access time of the CD-ROM player and its low transfer-rate, navigation within the data base through colour images, with the sole aim of reaching the information required, can be very slow.
It is therefore possible to navigate through the information in a black and white mode and activate colour only once the destination has been reached.
In order to take full advantage of all the potential offered, the following are necessary: a Macintosh II, with colour monitor and graphic card, and at least 4 Mbyte of RAM, mainly because the presence of colour images and animated sequences occupies a large quantity of memory. As all the images are also available in black and white, it is also possible to consult most of the data base with a Macintosh SE with 2 Mbyte of RAM, which is a much more common configuration.

2.2 Texts

Textual information has been considered an element of support which is necessary but not one of priority. It has in fact been preferred to leave the task of transmitting the information to the images, animated sequences and audio. However, the user has the faculty, if he/she so

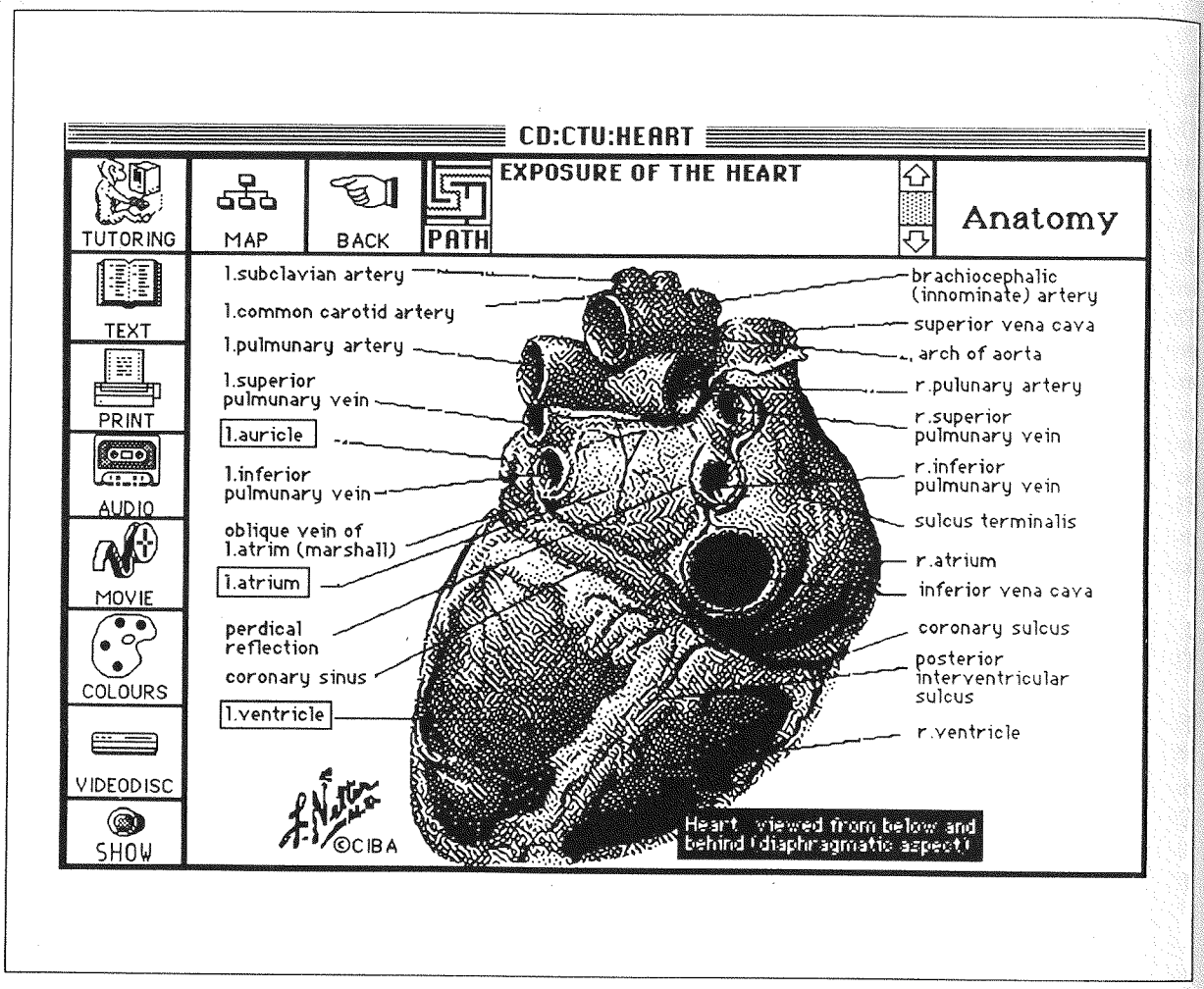


Figure 1

wishes, to read the text accompanying the images instead of listening to the commentary. It is possible to recall the images to which the text refers from the text itself.
It is also possible to have a print-out of the texts of the subjects available for consultation, which is automatically illustrated by the main image of reference.

2.3 Animated sequences

Some animated sequences have been created to illustrate the movement of the cardiac muscle during the systolic and dyastolic phases. The audio commentary associated with the animated sequences refers to the phenomena connected with the cardiac muscular activity and circulation of the blood and is synchronized with the images.

2.4 Audio

A commentary accompanies all the images, both still or animated sequences.
This enables the user to listen to the description of what he can simultaneously see in detail on the monitor.
A number of typical murmurs of various cardiac malformations are present in pathology and can be controlled interactively by the user.

3. NAVIGATING IN "THE HEART: ITS FUNCTIONS AND DISFUNCTIONS"

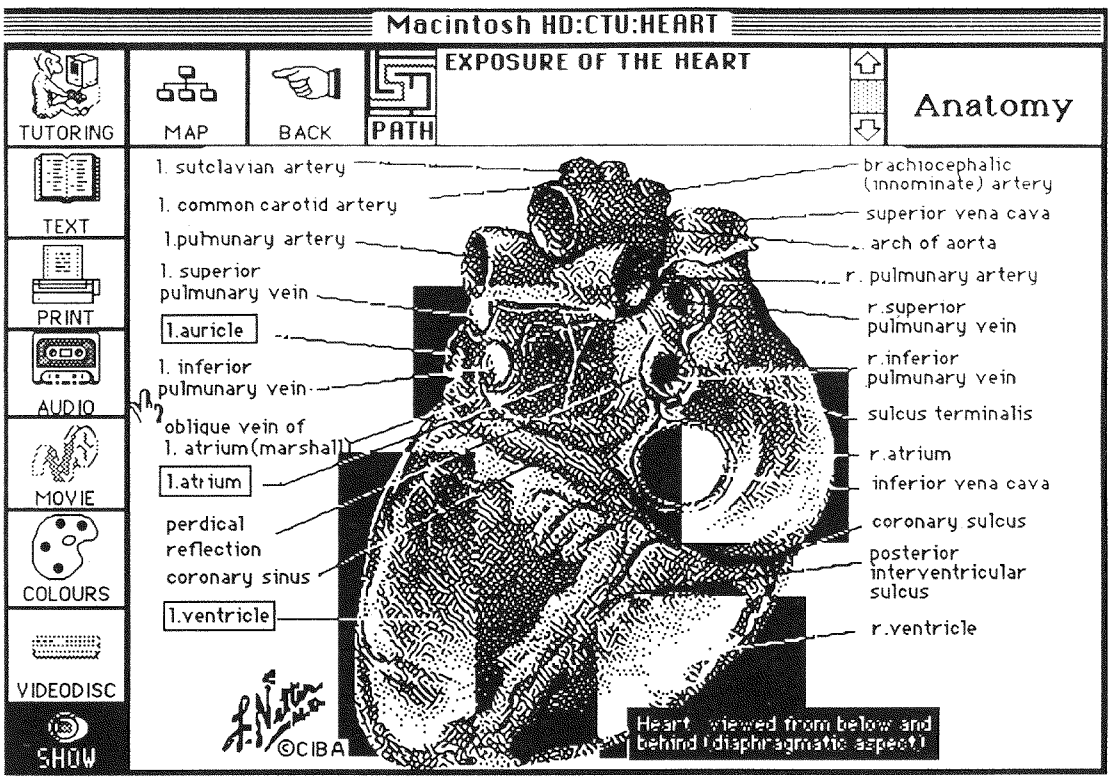


Figure 2

The simplest way of explaining the project is perhaps that of simulating a hypothetical navigation in its contents.

After a short introduction, the user is positioned on the image.shown in fig1

In order to understand the active links, the command "SHOW" is available.

Let us activate the button clicking on the mouse: some areas are highlighted (fig 2). When the mouse is released, the image reacquires its initial appearance.

It is possible to select one of the highlighted areas. Let us select the area corresponding to the "Right Atrium" (fig 3).

All the images of the database are present both in colour and in black and white. Let us activate the "COLOURS" button.

Through the activation of the "TEXT" command, it is possible to gain access to the text relative to the image being consulted (fig 4).

It was decided to use relatively large and well defined characters in order to try and facilitate reading on the monitor, which can be a tiring operation. For the same reasons, it was also decided to allow the user to browse through the pages one at a time instead of using scrollbars. The text can also be printed with the "PRINT" command. The main illustration to which the text refers and subsequently the whole text is printed, made up of the union of all the pages in the section. The "PRINT"

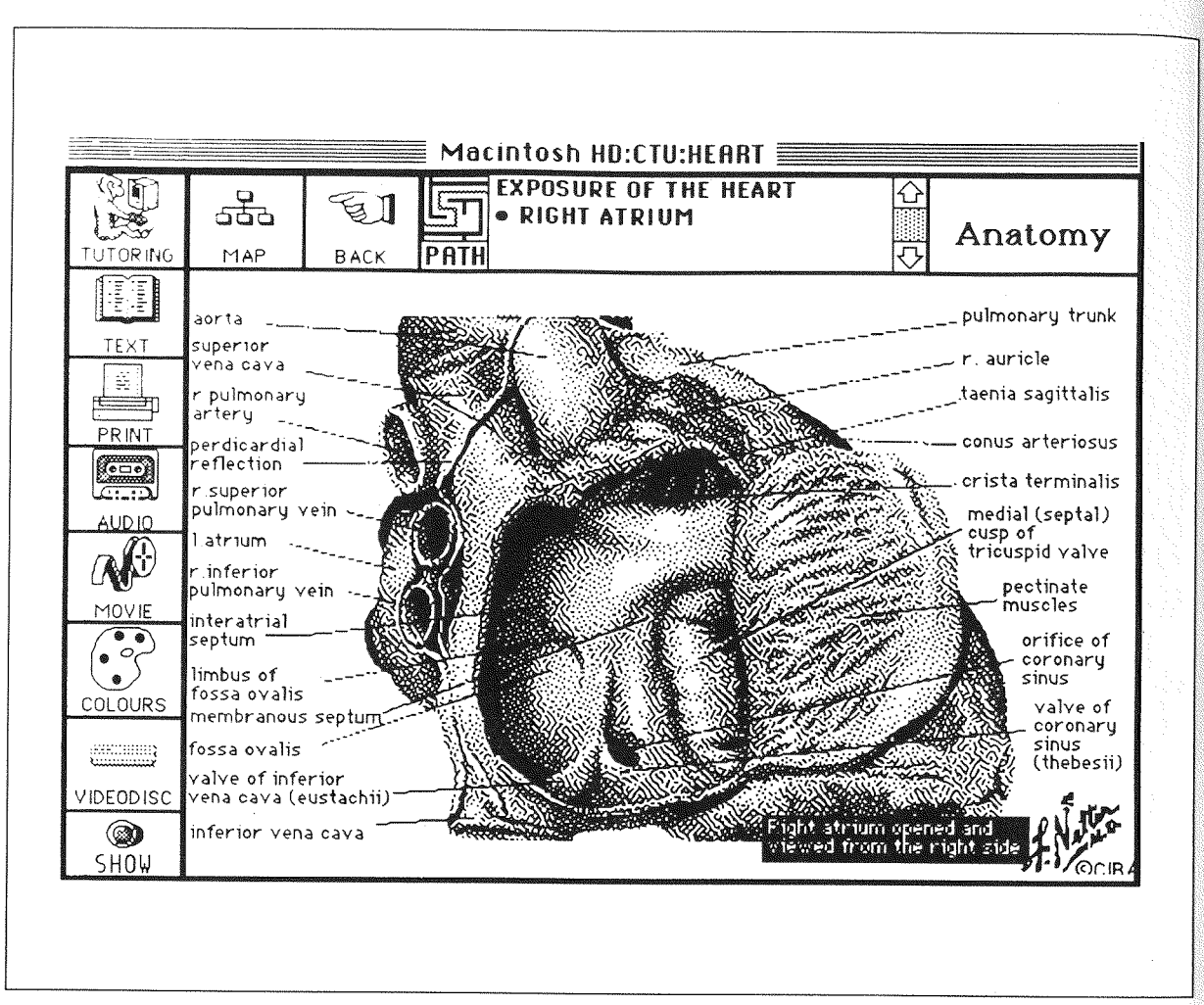


Figure 3

command is also present for the images (black and white only) in this case it produces the print of the image displayed at that time on the monitor.

Through the text it is possible to navigate inside the data base. If references exist to the images present in the data base, an icon appears in correspondence with the line of text where such a reference is present. If the icon is activated, it allows access to the section relative to the subject quoted.

With the "IMAGES" button, let us now return to the previous image (fig 3). Let us activate the AUDIO command; a pop-up menu is offered to us (fig.5): selecting the option PLAY, it is possible, while the image remains on the screen, to listen to the information shown in a textual form in the previous card.

Clicking the button at the top right (fig. 6) we go from "anatomy", now being consulted, to "pathology". If we choose the AUDIO button, we now have the recording to the murmurs of a number of cardiac pathologies. In other words, the four sections, anatomy, pathology, embryology and physiology, activate a grid of consultation of the data base, so that the interface buttons make different information available according to the section we are in.

After having listened to the audio, which can be suspended and begun again at will, let us activate the ZOOM command. Here again a pop-up menu (fig 7) offers us the possible alternatives of enlarging the image. Choosing one of the possible options, a scrolling window appears on the monitor, showing the image with the

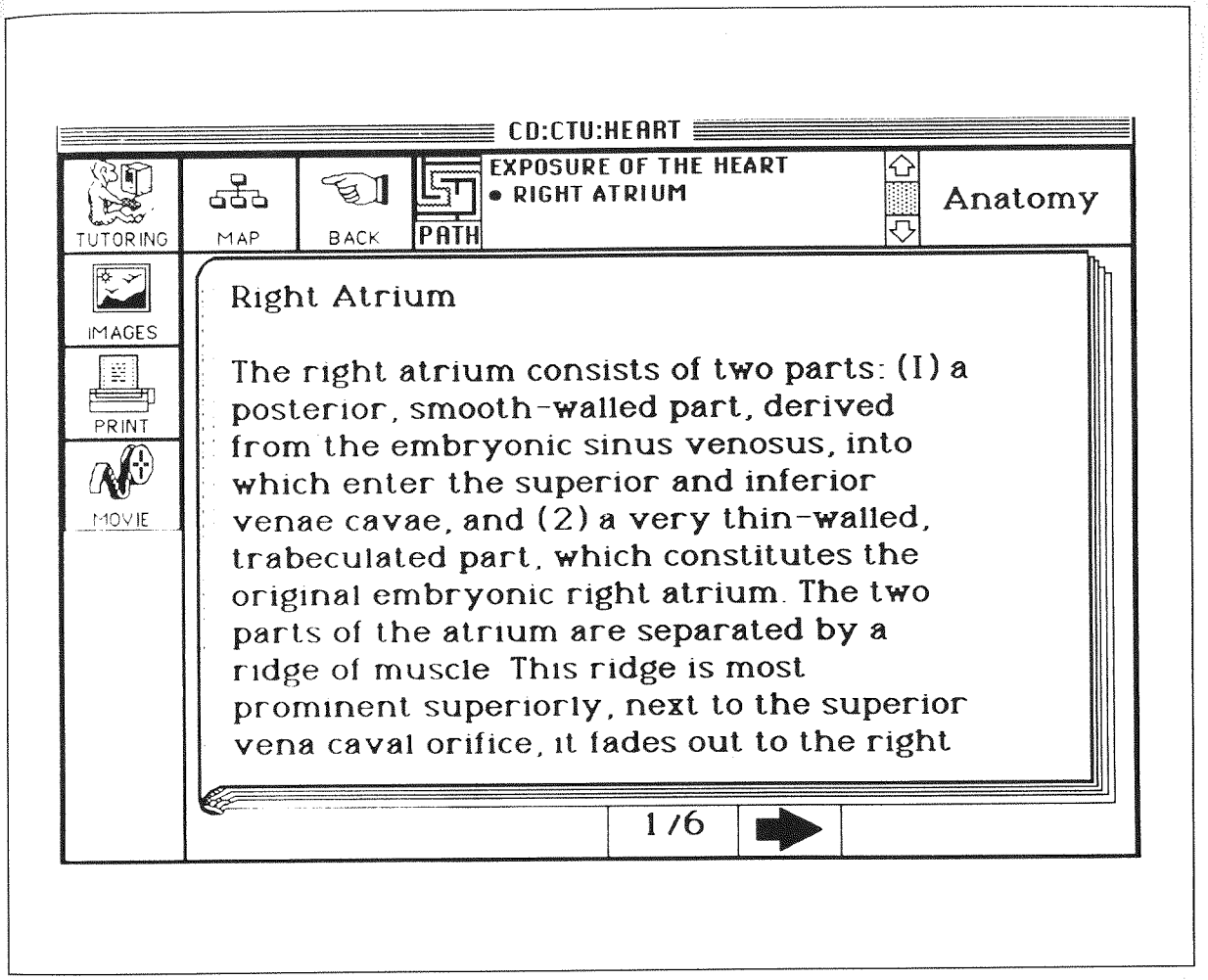


Figure 4

factor of enlargement requested by the user.
Let us close the window in which the enlarged image has been displayed to return to the image in a 1:1 ratio with the original.
Let us activate the MOVIE command (fig. 8) and select MOTION 2. Through the panel located on the left part of the card (fig. 9), it is possible to control the animated sequence and the associated audio commentary.
To return to the consultation of the data base, we have to activate the BROWSING command.
With MAP we go to the map, where a black arrow marks the point from which we have arrived, namely the subject being consulted. Choosing "Conduction System" from the Anatomy section, we are taken to a new section of the map. This represents the subjects dealt with in the section dedicated to the conduction system.
Let us now select the icon named SA-NODE and we are now once again in the data base.
Up to this moment, from the beginning of our work session, we have navigated inside the browsing environment. Let us see, then, what happens if we activate the TUTORING button. Fig. 10 represents what is displayed on the monitor.
We can note that the interface, going to this second mode of consultation, has changed. The subjects that we ought to know as pre-requisites of the section requested, are shown. We can choose one of them, more than one, or all of them, and confirm our choice with

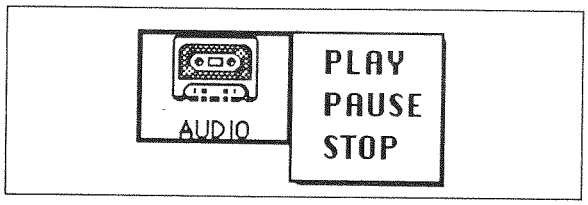


Figure 5

OK. We obtain the image presented in fig. 11
An actual lesson can begin from this point, like those obtained from a videotape. With the PLAY and PAUSE commands we can activate or suspend the lesson which is offered to us with sequence of still images and an audio commentary. To return to the browsing environment we use the relative command. The programme takes us back to the section of the data base corresponding to the subject in consultation in the tutoring mode. At this point it is possible to start the navigation again through the multimedia data base, activating the commands available through the programme, as the fancy takes us.

4. TOOLS FOR "FINDING THE WAY AGAIN"

A great deal of the literature on hypermedia systems underlines the problems of disorientation and getting lost in the hyperspace of information.
The problem of disorientation is described as the non knowledge of *where* one is in the information space or the non knowledge of *how* to have access to something which in all probability is present. The problem of disorientation is thus linked to the non knowledge of the

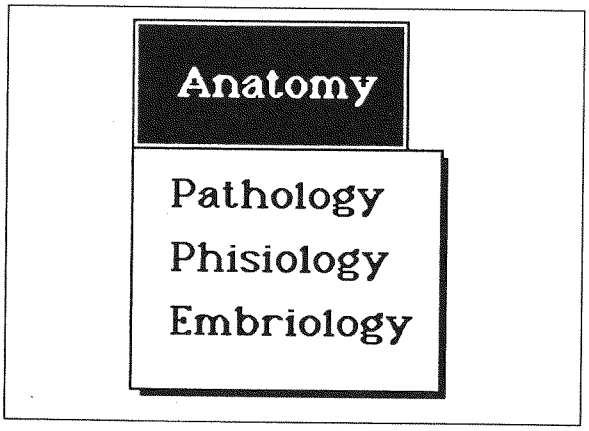


Figure 6

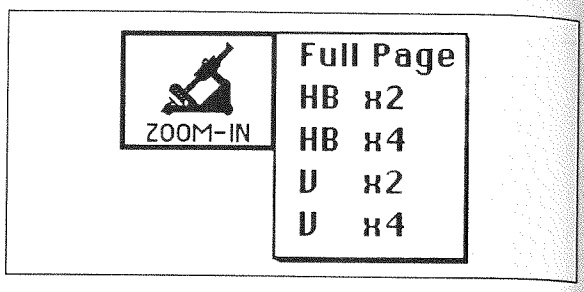


Figure 7

spatial arrangement of the nodes within the hypertext network.
All this is aggravated by the fact that the computer screen and resolution are limited. The fact of being able to have several overlapping windows allows the space available to be multiplied, but certainly does not aid orientation.
We have implemented some tools for "finding the way" in our programme.
Let us have a look at them.

4.1 History

A scrolling window is inserted, showing the position of the user with respect to the general data base.
In other words, the shortest path leading to the information in consultation is shown, through the hierarchical levels necessary to reach it only, even if the user has followed a much more complex path.
In the future implementation, this window will show the whole succession of units visited before the current one, dynamically updating this sequence at every significant command of the user.
Other functions which we consider fundamental exist. First of all, the history must be interactive. That is, it must be possible, clicking on the name of the information unit visited, to gain direct access to it, without passing through the map.
Furthermore, the history must in one way or another communicate the nesting of the subjects visited, that is

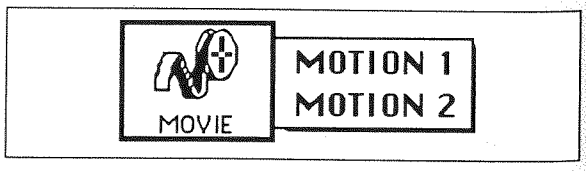


Figure 8

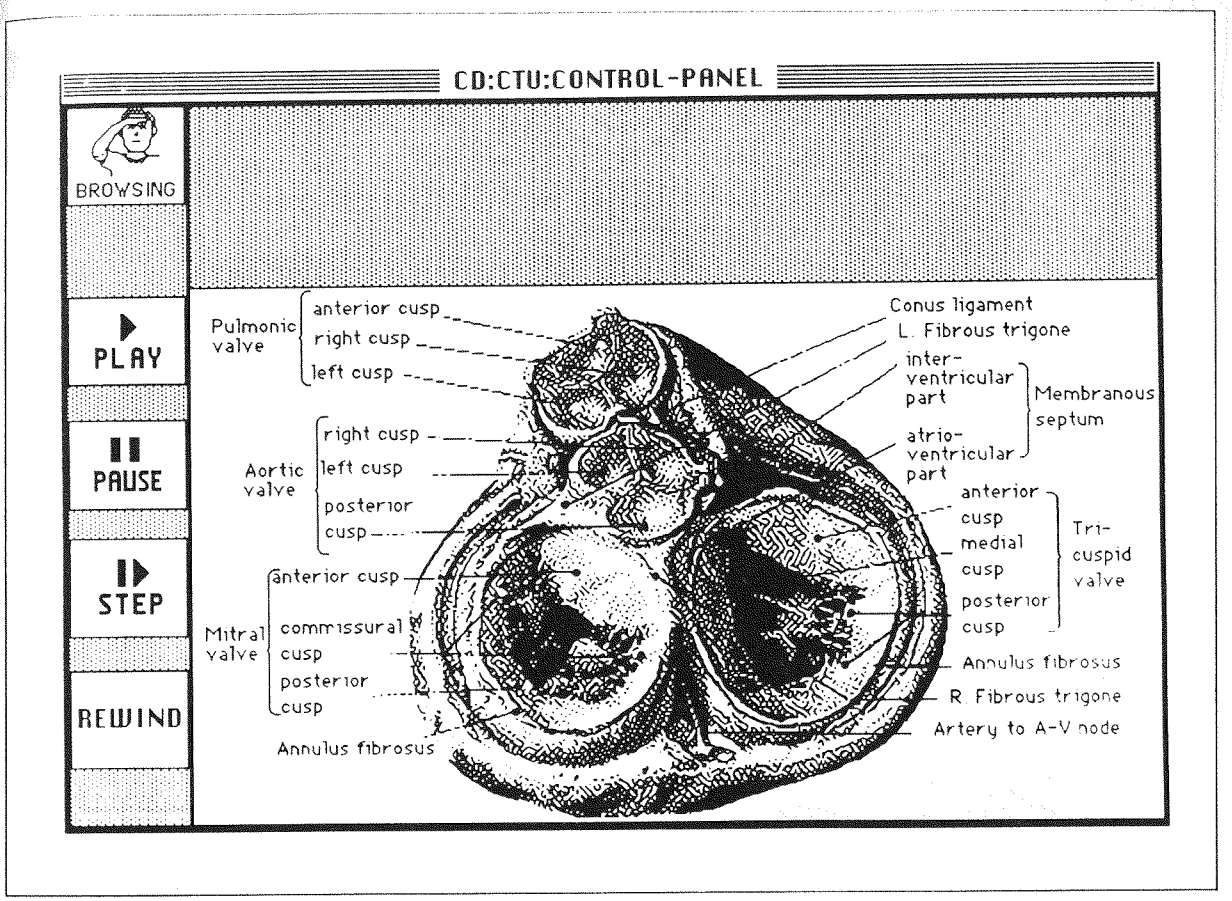


Figure 9

the hierarchical relationship with the other nodes, when it exists, and/or the position in the network when the connections are at the same level.
A history with icons could be useful, also to recall in graphic form the information explored.
Finally, it should be possible to save the history, or to build up a library of **personalized paths** which can be recalled at will. This obviously requires making an authoring environment available to the user, allowing him/her to easily take advantage of this possibility.

4.2 Back

A back command exists which allows the user to go back one step and retrace the whole path, one step back at a time.

4.3 Map

The map shows the relations existing between the system's documents.
A general map exists, illustrating the relations existing between the macro-contents. Some of the nodes of the map are buttons, that is, it is possible to obtain a more detailed view of the node in terms of other subnodes and links.
The black arrow indicates the position of the user inside the data base.
The map should have the possibility of supplying local views of the nodes and links of the data base, modified according to the user's position. This last possibility has not been implemented in our work.

4.4 Bookmarks

Like classic bookmarks, these indicate a piece of information that has been visited, in order to be able to find

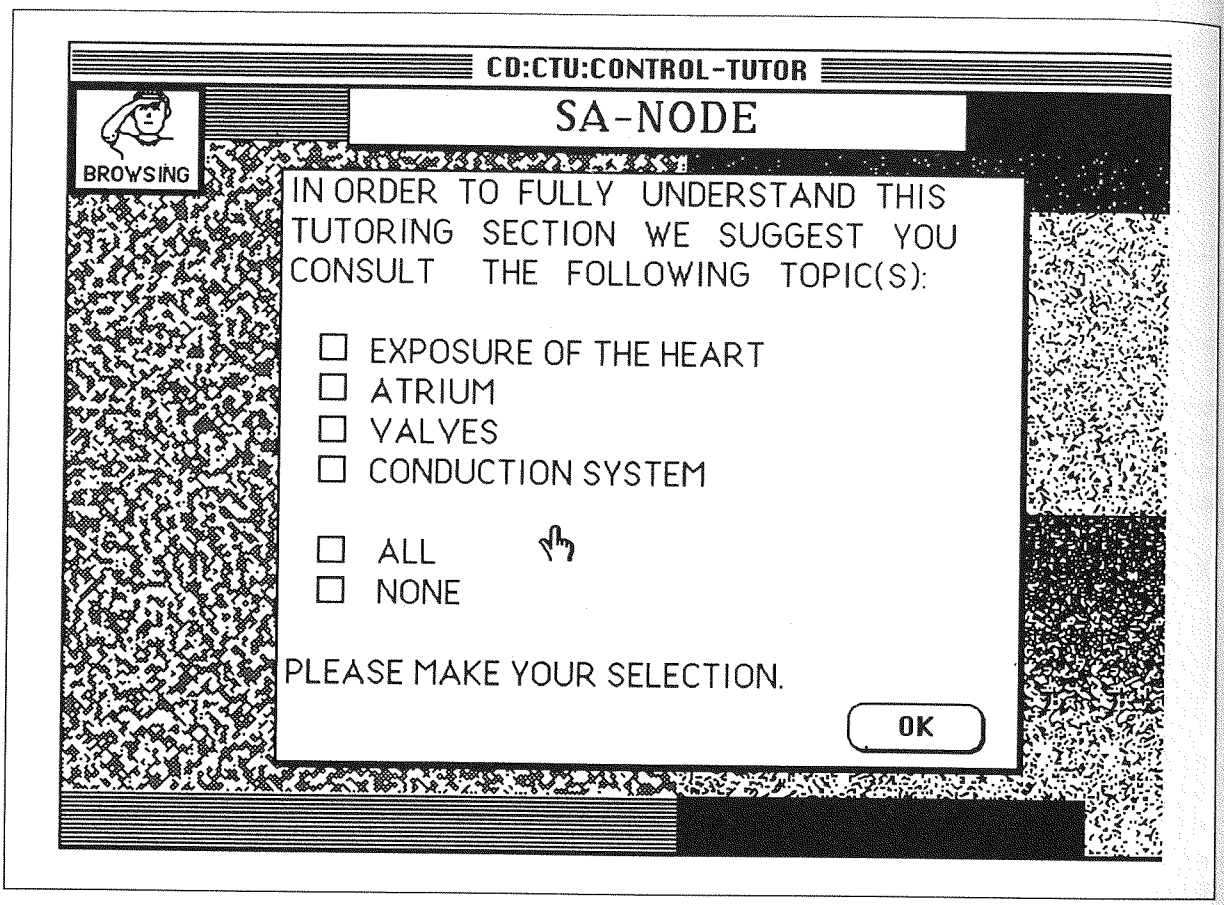


Figure 10

it again immediately, if so desired. They normally allow access to a path from other points, as well as from the beginning. We have not implemented this tool in the hypermedia.

5. THE INNOVATORY ASPECTS OF "THE HEART: ITS FUNCTIONS AND DISFUNCTIONS"

From the technological point of view, the realization of the project meant tackling several problems, the solutions to which were, at times, highly innovatory. Let us consider them.

5.1 Audio CD-DA on two mono channels

After several evaluations of the characteristics of the audio formats that could be used, it was decided to record the information in the CD Digital Audio format (Red Book). This format represents the standard supported by multimode players for CD-ROM. The reasons that led us to this choice can be summarized as follows:

I) Possible parallelism in the execution of the operations.

The PCM processor dedicated to the decoding of the audio files and resident on the CD player, is independent of the CPU of the personal computer being used. This allows a real parallelism in the execution of the operations, on condition that the CPU does not require access to the data recorded on the CD-ROM whilst this is being used to play the audio.

II) No RAM required.

None of the RAM is occupied for the audio decoding

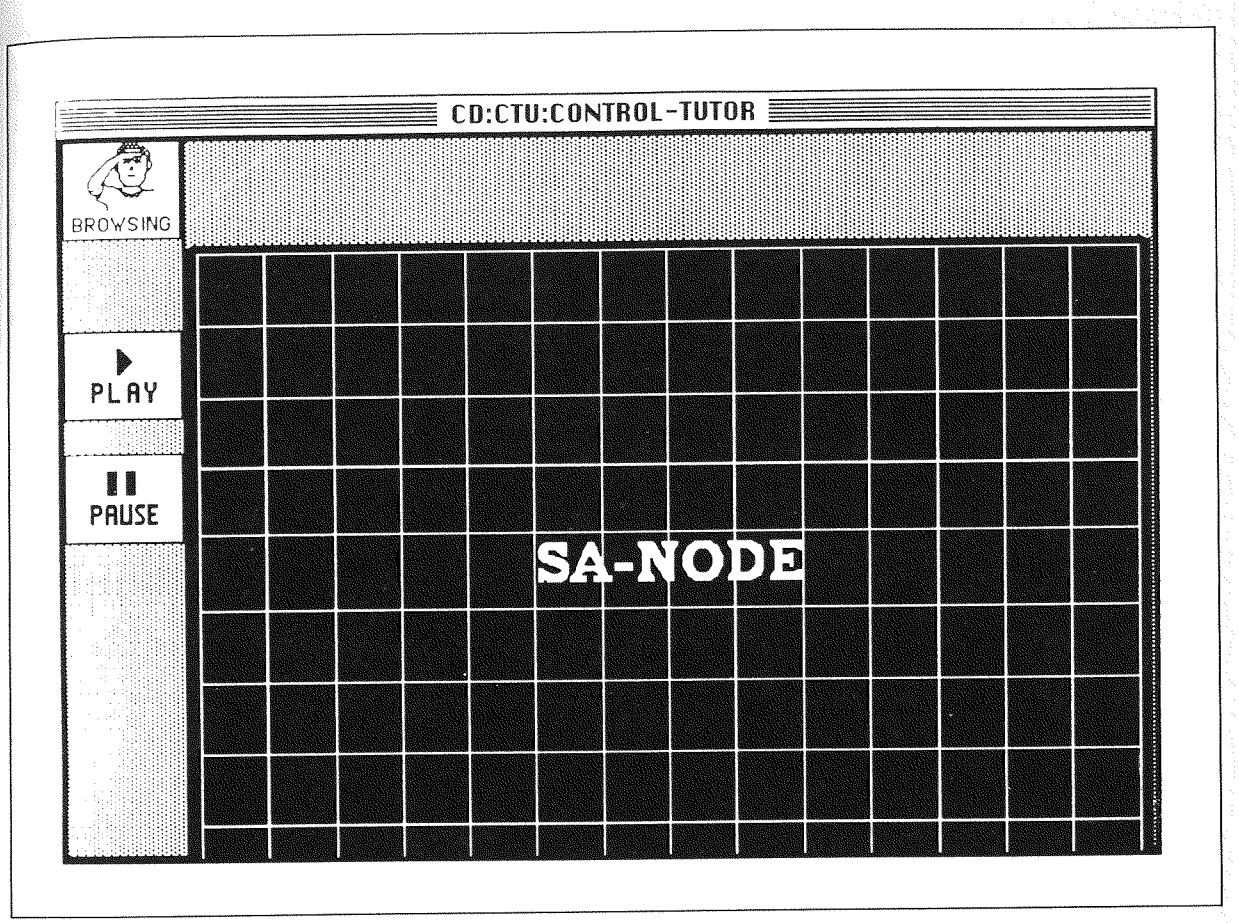


Figure 11

operations. The PCM processor carries out the "digital to analog" conversion in real time directly from the disc. In this case, therefore, the limitations on the length of the audio cuts to be reproduced depend on the physical space available on the CD-ROM and no longer on the RAM.

III) Addressing accuracy.

It is possible to address the player to any point whatsoever of an audio cut, with an accuracy of 75th of a second.

The major problem caused by having adopted the CD-Audio format is the large amount of memory occupied on the disc, about 172 KByte per second. This value, however, is relative to the recording on two STEREO channels which in our case (only spoken commentary) led to an excessive quality of reproduction.

If the original source of the information is in MONO, as is our case, the two channels are the exact copy of one another, and the same information must therefore be duplicated.

In producing the CD-ROM we have consequently exploited the two STEREO channels as though they were independent.

Recording different information on the two channels, it is possible to optimize the amount of memory occupied on the disc. This method in fact allows up to 144 minutes of MONO audio to be recorded on a single CD-ROM.

This procedure, however, entails a greater complexity in designing the application as well as in developing its software.

5.2 Synchronization of audio/video cuts

As well as a commentary to the still images, the audio has also been used in association with sequences of still

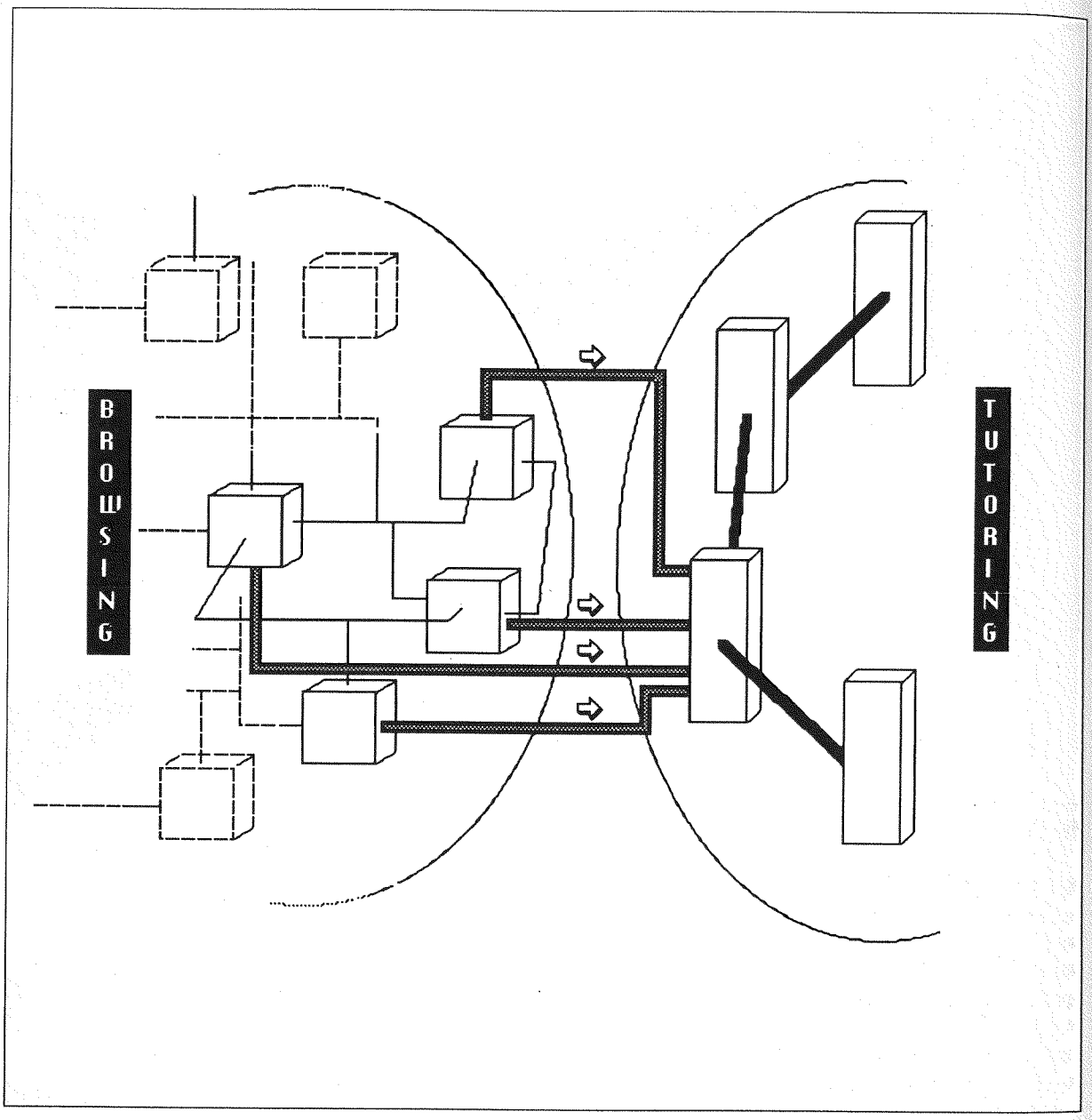


Figure 12

images (slide show, in tutoring) and with animated sequences. In the latter case, the problem was even more complex. It was decided to produce some animated sequences with a synchronized audio commentary which could be controlled interactively by the user. This led to the production of a Control Panel for the animated sequences, which allowed the user to carry out the operations of PLAY, PAUSE, STEP and REWIND. This was possible thanks to some software routines which allowed the file of animated sequences to be treated as a list of frames which could be addressed individually via the software.

6. BROWSING AND TUTORING

Navigation in search of the required information through a wide set of data using hypertextual paradigms can be a good approach to finding the information.
As far as navigation through the information is concerned, our approach presents some innovatory aspects with respect to standard hypertextual systems.
In the CD-ROM "The Heart: its functions and disfunctions" two possible modes of consultation are possible: BROWSING, the classic mode of hypertexts, and TUTORING. We have considered the latter mode, which is seldom found in hypertextual applications, essential in order to meet the didactic goals of the program. The BROWSING mode guarantees that the user has complete freedom, with the possibility of building up a personalized path through the contents. This is the mode in which the user is at the beginning of a work session.
In this mode, which has a hypermedia architecture, all the tools necessary to move through the information are supplied, but no predefined paths exist.
The tools offered by the interface in the BROWSING mode are dynamically connected with the environment.
In the TUTORING environment a consultation is suggested which has the aim of reaching pre-established didactic goals. In this mode, the user is offered "lessons" on the subjects previously selected. The program lets the user konw, at the beginning of each tutoring session, which kind of subject he/she has have mastered to be able to fully understand the lesson.
The lessons are made up of sequences of images with a synchronized audio commentary. On each image the parts commented upon by the speaker are highlighted in turn.

A very interesting aspect is that the TUTORING modules are contextual with respect to the BROWSING mode. When a lesson in TUTORING is interrupted, BROWSING is entered at the point where the TUTORING session has been interrupted. In the same way, when TUTORING is accessed from BROWSING, the user is automatically inserted in the module on the subjects being explored. However, the passage from BROWSING to TUTORING is not as simple as vice versa, i.e. from TUTORING to BROWSING. TUTORING differs from BROWSING from the didactic point of view, as at the end of each module it is supposed that the user has acquired a series of skills and that in the subsequent module these skills are considered acquired. This is obviously not possible in BROWSING, where the user is left completely free to explore the contents and where normally extemporaneous links are followed rather than a rigorous logical sequence. Consequently, the skills that the user is expected to have or in other terms, the prerequisites for the module about to be consulted, are shown at the entrance to each TUTORING module.
The relation existing between the information units contained in the BROWSING mode and the TUTORING modules is of a "many to one" type, as shown in the diagram (fig. 12).
The diagram shows how the various transitions from BROWSING to TUTORING are made. The arrows indicate the direction of the path.
When the user makes the request using the appropriate command to go to the TUTORING mode, the program first identifies the information unit and environment currently being consulted in BROWSING; transition then takes place to the TUTORING module in which the subject is treated.
It is obvious that all the possible links have been predefined and make up part of the information associated to each individual unit of information.
The inverse relationship, namely between the TUTORING modules and the BROWSING units, is of a "one to one" type. A main subject exists for each lesson in TUTORING. The information unit corresponding to this main subject represents the entry-point to the BROWSING mode.

7. LESSON LEARNED AND FUTURE DEVELOPMENTS

A number of points have emerged from the creation of "The Heart: its functions and disfunctions" program, first of all from the methodological point of view.
Our goal was completely new and it was difficult to find an appropriate model for communicating between the designers and the experts on the subject.
"The Heart: its functions and disfunctions", which has translated an idea into a product, has simplified our approach to the university teaching staff with whom we collaborate as the program gives a practical demonstration of the limitations and possibilities of the technolo-

gy. The object cannot effectively be described by words, and even less by a paper.
We also now have a clear idea of the costs and times required for a product of this kind.
A further problem to be faced is the lack of design and production methodologies fitting the technological environment and the model of navigation which we wanted to implement.
One of the results of "The Heart: its functions and disfunctions" has also been the definition of a possibile methodology of work.
In the meantime, software tools for design and production have appeared on the market, which will simplify our work for the forthcoming titles.
In order to create a hypermedia product, we then saw that it was essential to have software which could integrate and process different types of information.
Other important considerations relate to the software for the hypermedia applications. The complexity of these applications requires the use of sophisticated software tools dedicated to this purpose, in order to avoid developing a large amount of software in programming languages such as C, Pascal or Assembler.
The ideal situation consists in having available software and a high level programming language supplying a set (as wide as possible) of predefined procedures for:

- processing of multimedia data (text, images, audio)
- the control of external devices such as CD-ROM or video-disc players
- the creation and control of hypertextual structures
- the creation of graphic interfaces and the management of windows, pop-up menus, buttons etc.

In addition, it is fundamental to have the possibility of expanding the set of predefined functions and procedures according to the user's individual requirements. This should be possible through the production of software modules written in the commonest programming languages, which can be recalled by the application at the appropriate time. The hypermedia software should thus allow:

- parameter passing from and to the external procedure
- maintaining the environment (state of the variable and screen interface) in which the application is located at the time of the recall
- communication between the application and the external module during the execution of the latter.

If the software used for the multimedia application does not respect the parameters as mentioned above, it is necessary to develop a lot of complex software routines in order to solve each specific problem, increasing the complexity of the program and greatly decreasing productivity.
From the technological point of view, it should be pointed out that today digital multimedia on PC is still at a very primitive stage. Or we can say that there are two aspects of the problem: video professionals find it frustrating when trying to produce applications that use a great deal of images and audio in a digital environment. On the other hand, computers professionals find the possibilities offered today by multimedia exciting. However, when it is possible to combine the treatment of multimedia on analogical and digital supports, a whole new world of possibilties is opened up.
The treatment of moving images, which is very simple on videodiscs, can be accompanied by a large amount of audio, now possible on CD-ROM XA and still images with a degree of resolution and stability (that is, without any flickering which is a classic disturbance in an analogical environment) never before attained.
This idea of combining analog and digital brings with it, however, very high costs for delivery systems.
CD-I and DVI could represent a solution, but it will be crucial, for the possibility of carrying out routine productions, to have authoring tools which support, in these environments, all the phases of production, from design to simulation. Some of these tools were seen at the last conference on CD-ROM held in San Francisco at the end of last February.
From the didactic and communicative point of view, digital multimedia has opened up exciting prospects offering the final user the possibility of manipulating the data.
This means that there is the possibility of a real exchange of roles between the author and the user, the latter passing from an attitude of spectator to that of actor.
The concrete possibility exists that the person using the program suggests a new navigation through the data, or their modification, in a sort of work of cooperation which, stimulating an active participation in the construction of the meanings, allows effective learning.
"Computers are not just for computing!
We are beginning to use computers, videodiscs and CD-ROMs to communicate in new ways. Computers are not

merely for manipulating numbers in a spreadsheet or words in a word processor; they can be used to access, manipulate and display information in many forms (including motion pictures, animations, music and sound effects.). It is the combination of these forms that will evolve into a new medium of expression-the multimedia composition" (Sueann Ambron, 1990)

8. REFERENCES

[1] AA.VV., NAVIGARE CON GLI IPERTESTI, Allegato a ZEROUNO n.95 del Dicembre, Mondadori Informatica, 1989

[2] Ambron, S., Hopper K., LEARNING WITH INTERACTIVE MULTIMEDIA. DEVELOPING AND USING MULTIMEDIA TOOLS IN EDUCATION., Washington, Microsoft Press, 1990

[3] Apple, HUMAN INTERFACE GUIDELINES: THE APPLE DESKTOP INTERFACE, Addison-Wesley Publishing Company, Inc, 1987

[4] A. Campana, UNA METODOLOGIA PER LA RAPPRESENTAZIONE UNIFORME DI ARCHITETTURE IPERMEDIALI. L'ESPERIENZA DEL CD-ROM "THE HEART: ITS FUNCTIONS AND DISFUNCTION", Tesi di Laurea in Scienze dell'Informazione. A.A. 1988-1989

[5] R. Cattaneo, INTEGRAZIONE DI MEDIA DIVERSI NELLA MULTIMEDIALITÀ DIGITALE: METODOLOGIE, TECNOLOGIE E REALIZZAZIONI. L'ESPERIENZA DEL CD-ROM: "THE HEART: ITS FUNCTIONS AND DISFUNCTIONS", Tesi di Laurea in Scienze dell'Informazione. A.A. 1988-1989

[6] Marlin "Lin" Brown, C., HUMAN COMPUTER INTERFACE DESIGN GUIDELINES, New Jersey: Ablex Publishing Corporation, 1988

HYPERMEDIA SECRETARY FOR DOCUMENT MANAGEMENT (*)

Piera Piardi
Dipartimento di Scienze della Informazione
Università degli Studi di Milano

RIASSUNTO

Il presente articolo descrive S2, un prototipo realizzato nell'ambito dell'automazione d'ufficio, utilizzando sistemi ipermediali. Tale prototipo rappresenta, attraverso un'interfaccia cognitiva, parti del mondo reale sulle quali l'utente può agire e fornisce meccanismi per l'esplorazione di questo mondo artificiale. Uno di questi è l'Agente. L'Agente opera in questo mondo ed esegue per l'utente operazioni specifiche su oggetti di questa realtà artificiale di cui è esperto. Il sistema realizzato S2, in cui l'utente dispone di oggetti a lui noti e di collaboratori, si propone di risolvere alcuni problemi relativi alla comunicazione, di tipo epistolare, in un'ufficio, offrendo sia criteri di classificazione di documenti, sia un insieme di funzionalità per lo svolgimento delle attività connesse.

ABSTRACT

This paper describes S2, a prototype created within the sphere of Office Automation utilizing Hypermedia Systems.
This prototype represents by means of a cognitive interface some real world fragments with which the user can interact in his usual way and also supplies interesting tools for the exploration of this simulated world.
One of these particular tools has been named agent.
The Agent operates in this world and executes, just as the user would, specific tasks on the objects in the simulated world of which it is expert.
S2, the system implemented , makes well known objects and collaborators, available to the user with the aim of resolving problems, such as office mailing, inherent to communication.
S2 offers documents classification criteria as well as a group of functionalities for the development of the relevant activities.

1. INTRODUCTION

S2 is the natural followup to a lot of research and activities developed and implemented at the Dipartimento di Scienze della Informazione of the University of Milano, in the sphere of Office Automation, all based on the Theme and Discipline concept.
It is widely known that some human activities can be more or less efficiently simulated by computers. However, in reference to Automation, one problem is to find out who or what activates mechanical activities.
Therefore it is important to analyze not only the events, such as the execution of more or less complex procedures, but also the conditions which cause the occurrence of specific events [2].
An office can be thought of as a communication system in which Theme and Discipline concepts result. Theme is the meaning of the communication and Discipline is the way this communication is carried out. The interac-

(*) This article represents a further development of the paper: "S2: Una segreteria cognitiva intelligente per la gestione di documenti di ufficio" (G. Caja, P. Piardi, C. Testero) presented at the A.I.C.A. conference "Text Processing IV", December 1989 [1].

tion between Theme and Discipline determines specific procedures activation. Since the communication elements in an office are conspicuous, it is very difficult to define the Theme. However by restricting our attention to the electronic communication of texts, we can - by recognizing a sequence of words having special meaning in specific contexts - locate the Themes.
This recognition will take place during the text editing. Actions (procedures) can be associated to every identified Theme: simply by writing his document, the user will activate procedures [3].
On the basis of the Communication Theme and Discipline concept one research project and some degree theses have been carried out at the Dipartimento di Scienze della Informazione.

The research [4], based on internal and external manager's correspondence, has allowed the following :

- to define a logic structure of documents
- to detect an elements classification based on the concepts expressed in the documents or based on words used to write them
- to define the associations between the recognized elements in a text and the executed actions (or those that will be executed).

Among others, the degree thesis entitled : "Il Sistema S" - [5] and [6] - has been of particular interest for the development of the project.
This thesis, done on a Unix system, rendered automatic

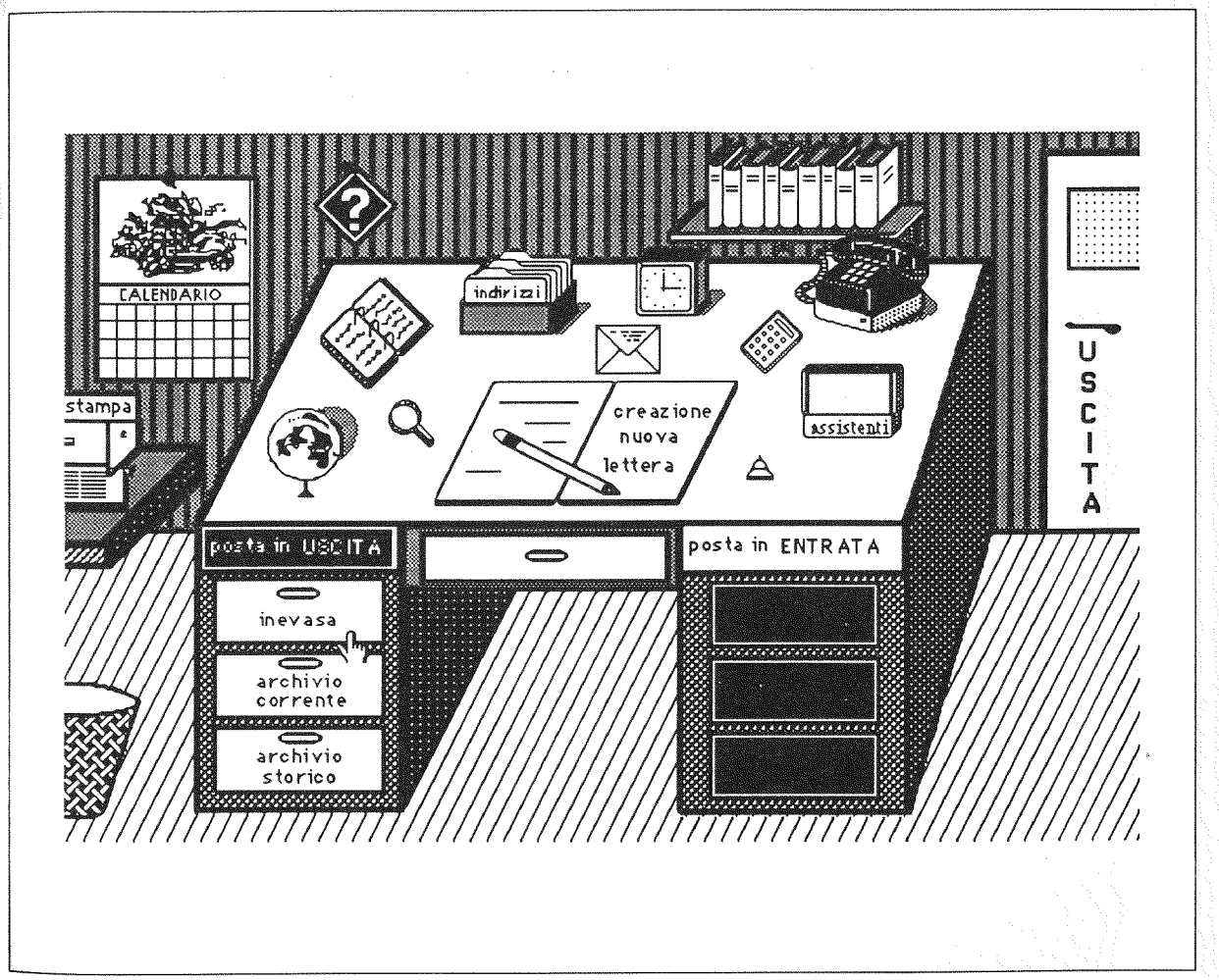


Figure 1: The Desk-top environment

the execution of procedures triggered by the recognition of meaningful words in a text.
The present work which follows the same philosophy, makes use of the advantages provided by the new technologies of hypermedia systems.
If we keep in consideration the Artificial Reality paradigm [7], the realized system is a tool that is able to model, through a metaphor, some aspects of office reality, using suitable multimedia representations of real environment and behaviors.
Using Hypercard, which allows appreciable graphic representations, we have been able to make an intelligent secretary system in which the user has at his disposal collaborators and known objects [8].

This kind of representation notably reduces the difficulties caused by the huge difference existing between the solidity of an objective, familiar real world and the abstractness of usual soft-packages.

An effective example of this difference reduction is provided by the representation on the screen of an office-room in which the user has, as in the real world, a writing-desk, a printer, a bookcase and many other objects at his disposal.
Every object is represented as it is in the real world, and its activation causes the execution of different procedures.
The Agent operates in this office.

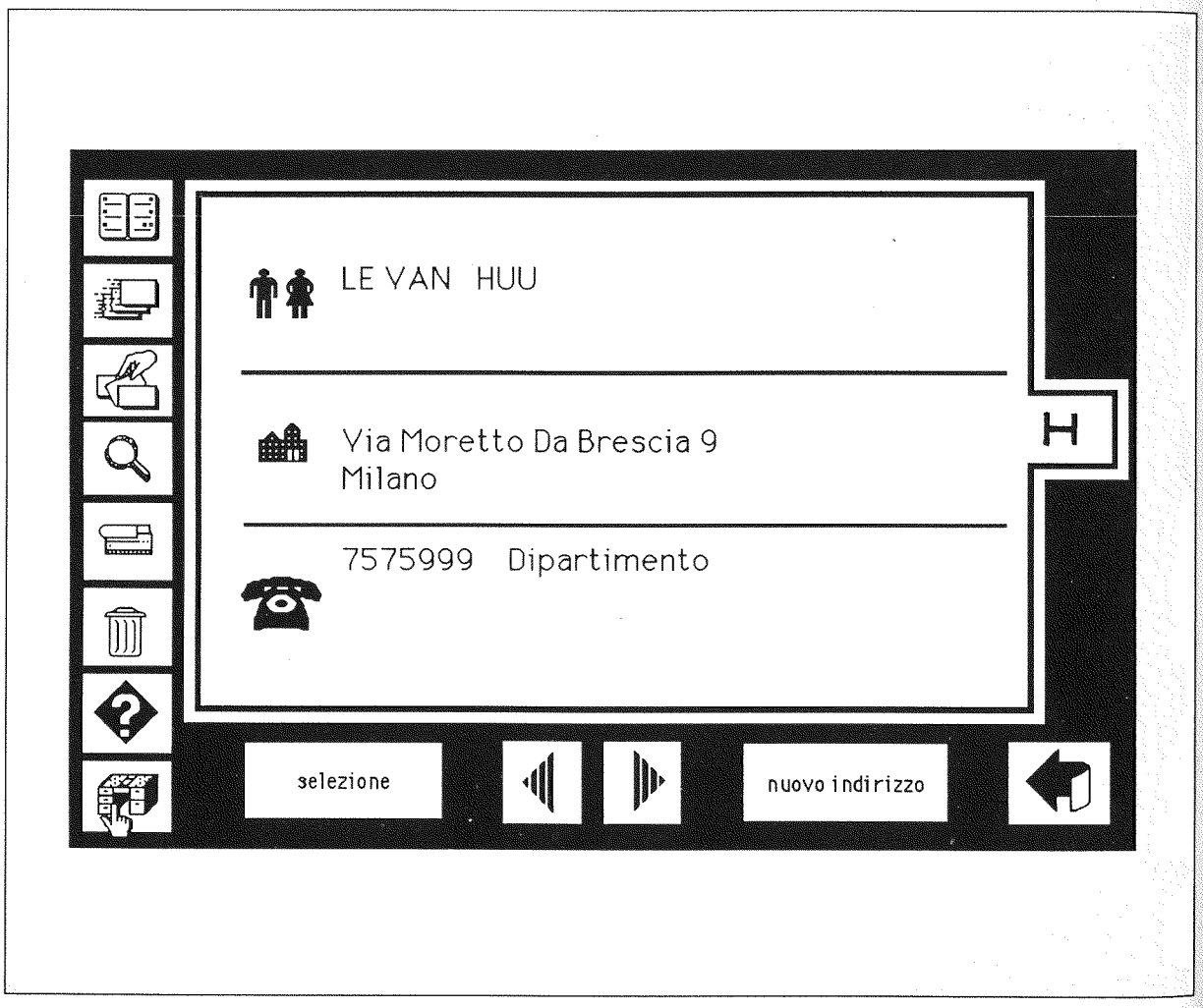


Figure 2: The Address Archive

2. THE AGENT

What is **"The Agent"** ?
The Agent is a program which according to the context in which it is working, performs different actions on the entities of which it is expert: such entities can be files, documents, information and so on.
Every Agent has its own particular knowledge and works only on the objects of which it is expert (otherwise it works on the objects that it knows). Therefore there are some different types of Agents characterized not only by the work that they have to do or by their world of knowledge, but also by the entities on which they do their work.
For example, let us assume we have an Agent whose job it is to retrieve letters throughout a selection of subject items criteria: this one surely will not be able to retrieve documents throughout addresses. Analogously, the Agent that has taken up transferring a letter from one archive to another, will not be able to identify the receiver of that letter. These Agents are all of the same kind and every one is called Elementary Agent.
The Agents employed on different jobs and objects can communicate together by exchanging messages.
The exchange of information and results allows the execution of a complex program, called by the compound of Agents.
Because the Agent works under an heuristic algorithm, it is necessary that the user always controls the situation confirming or cancelling the proposed activity or the

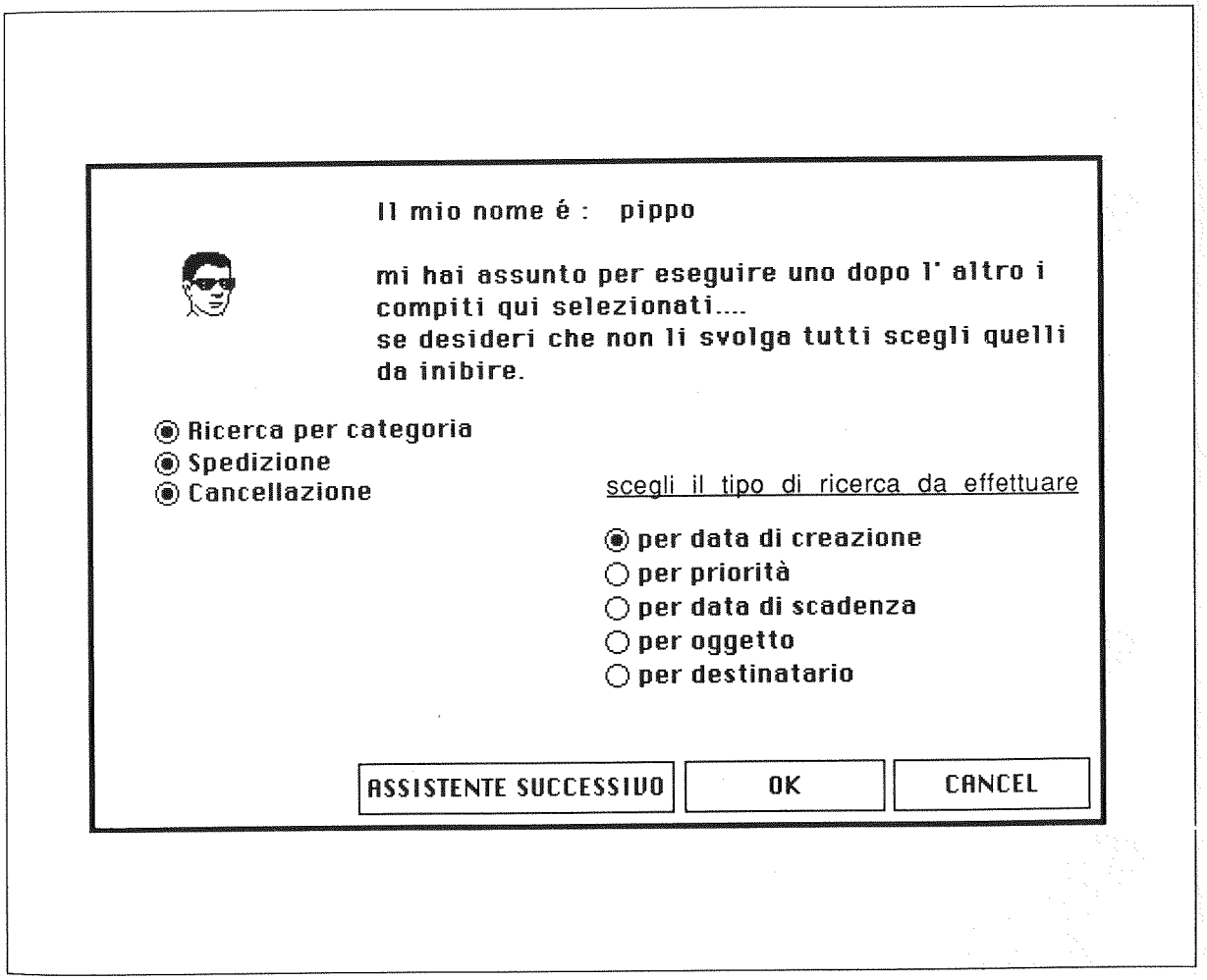


Figure 3: Assistants' Composition Environment

running activity.
The Agent may intervene upon a specific call of the user, who needs collaboration, or upon a system call that needs to communicate the activities that are to be developed or the consequences of operations already made by the user.
The Agent is characterized by an icon, representing a face and by a name that allows the unambiguous identification of that agent. This kind of representation allows the user to associate naturally and immediately the capacity for performing a specific job to a particular shape.
Finally the user can delegate to a specific Agent the development of the activities for which it has been accomplished, as in the real world it is possible to delegate specific functions to a co-worker.
In practice, the Agent can be considered as an assistant or a clerk, helping the user during the development of some of his activities [9].

3. THE PROJECT

The system is made up of 8 different contexts, every one of which is a node related to the others, following a hypertext structure [10] [11] [12].
Each context is characterized by the presence of objects related to the activities that the user can develop in that environment.
Those objects are Hypercard buttons: that is, they are screen-sensible areas associated to a script; otherwise,

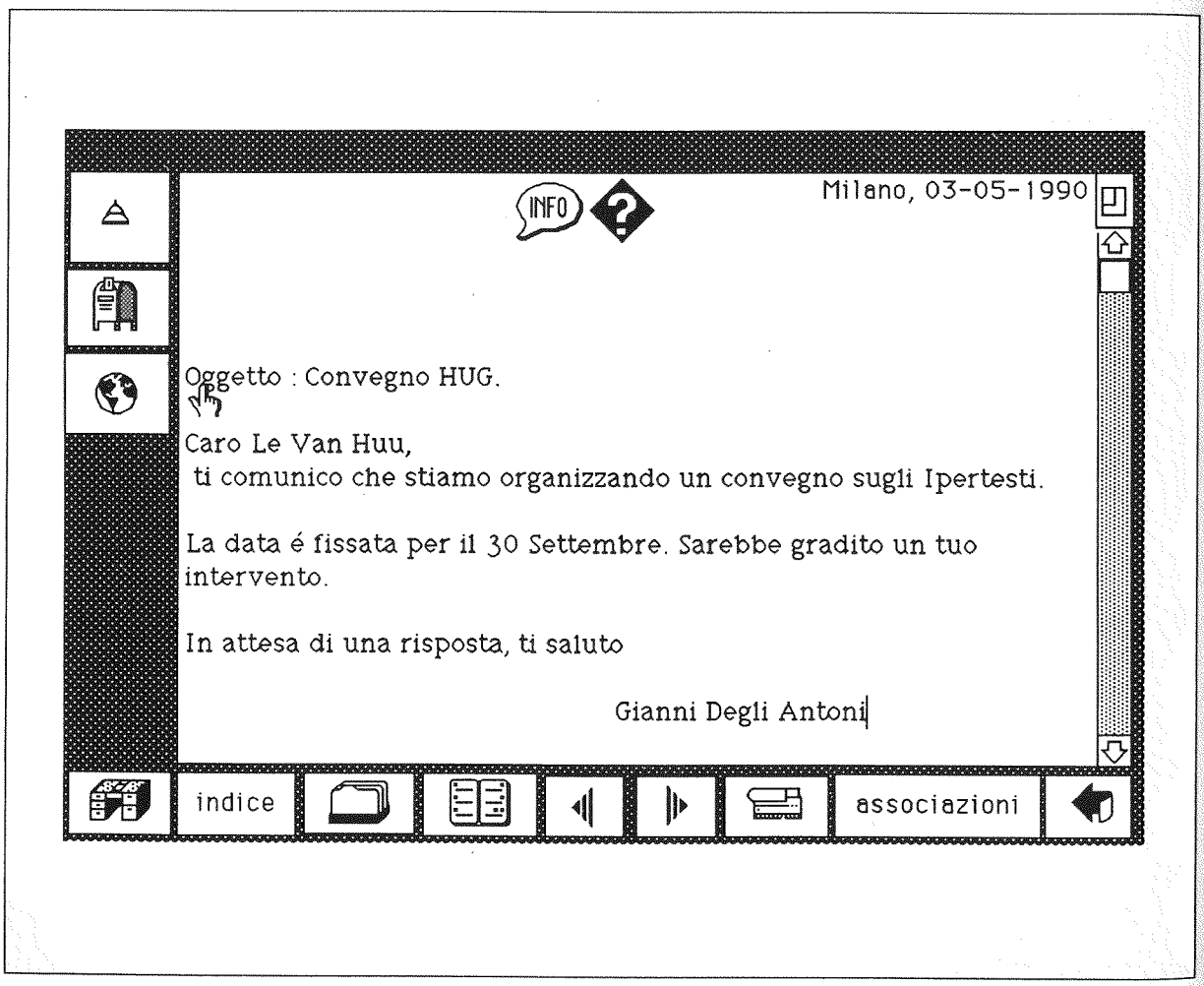


Figure 4: Documents Creation Environment

they are a program that performs actions when events occur.
Some buttons, being a link for navigation in the hypertext net, allow the user to display, at his wish, the other job's environments.
Other buttons perform more complex functions related to officeactivities. For the above it is evident that the buttons' functions differ due to the context in which they are activated.
The graphic representation has been carefully designed in order to avoid all possible ambiguity about the meaning of the operations that it's possible to do on behalf of the identified object. It can be done by an icon, selected so that it can be consistent with the real world, as in the case of Macintosh representation; otherwise, if a sufficiently expressive icon could not be found, a name can be used.
Here below we will examine the different environments realized in the present project: the buttons, the objects and the Agents.

4. THE DESK-TOP ENVIRONMENT

The ***Desk-top environment***, representing an office room on the screen, is the main one because from here the user has the global perception of all those in which he can perform actions.
Many of the real world objects are represented in this room; every object is a button that, if activated by a mouse or other, performs the function it represents.

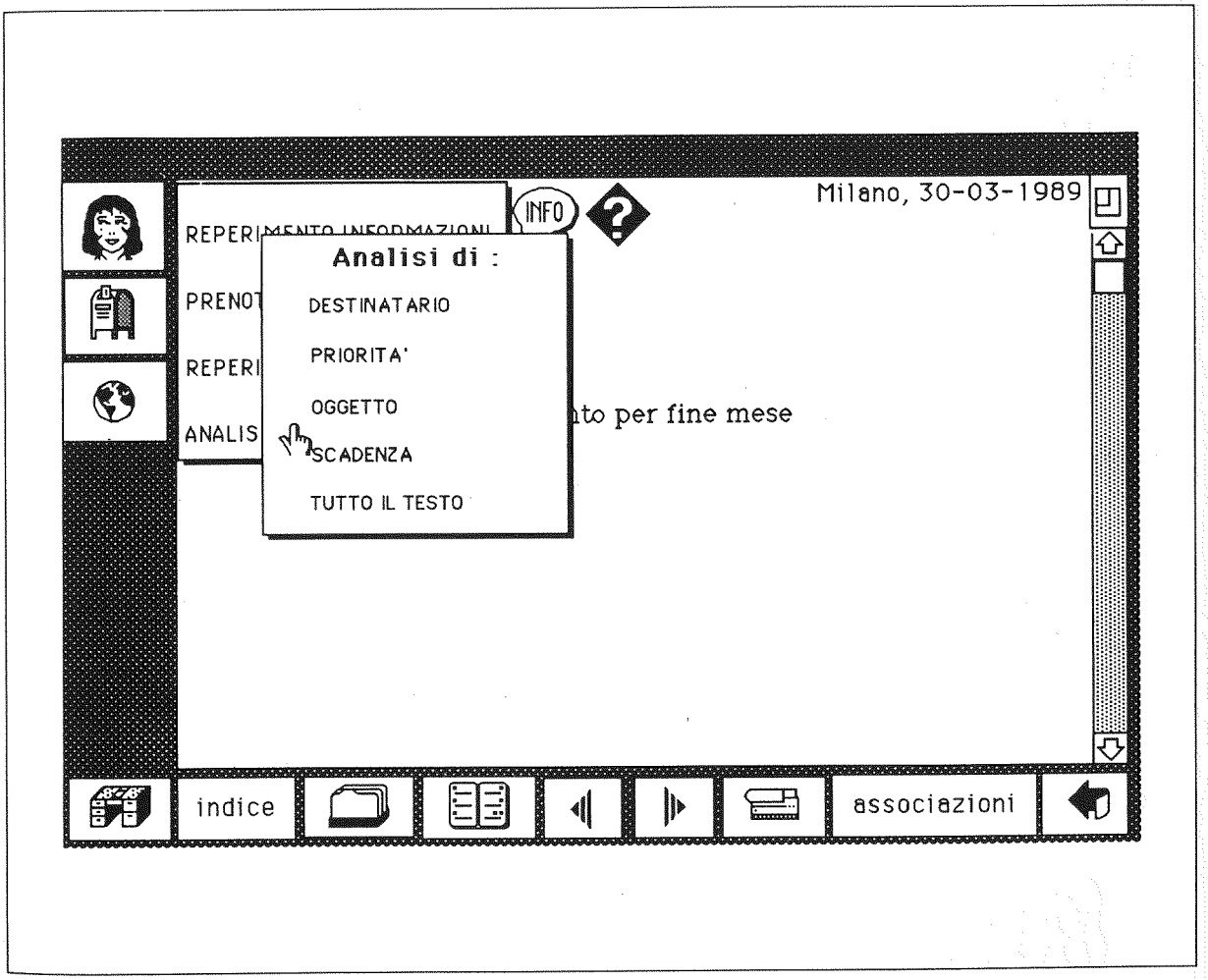


Figure 5: Analysis Function

For example, the ***Bookcase***, shaped as a bracket on the wall, allows the user to enter Hypercard external documents, whose headings correspond to the titles displayed. So it is possible to define not only the links in the system's hypertext net, but also the connection link between the system and external applications.
The ***Help***, represented by a Question mark, establishes a link with the environment in which all the system functions are explained, displaying not only the information related to the job's environment (into which the help has been called), but also the explanations of the last action performed in this environment.
Activating the ***Magnifying Lens***, the user can retrieve, using a specific key, a document from the system archives.

The retrieval of at least one letter allows the execution of the print procedure, the transfer from one archive to another, the cancellation of one or more documents.
The script associated to ***waste-basket*** executes the cancellation request only when it recognizes that a selection of one or more letters in the system archives has been done; otherwise, the script informs the user about the necessity of a retrieval activity on the archives.

The ***printer*** button works in the same way: the user is informed whether or not a document has been selected for printing. Opening the ***desk's drawers***, makes it possible to go to the corresponding archives and perform different actions on the documents, such as tran-

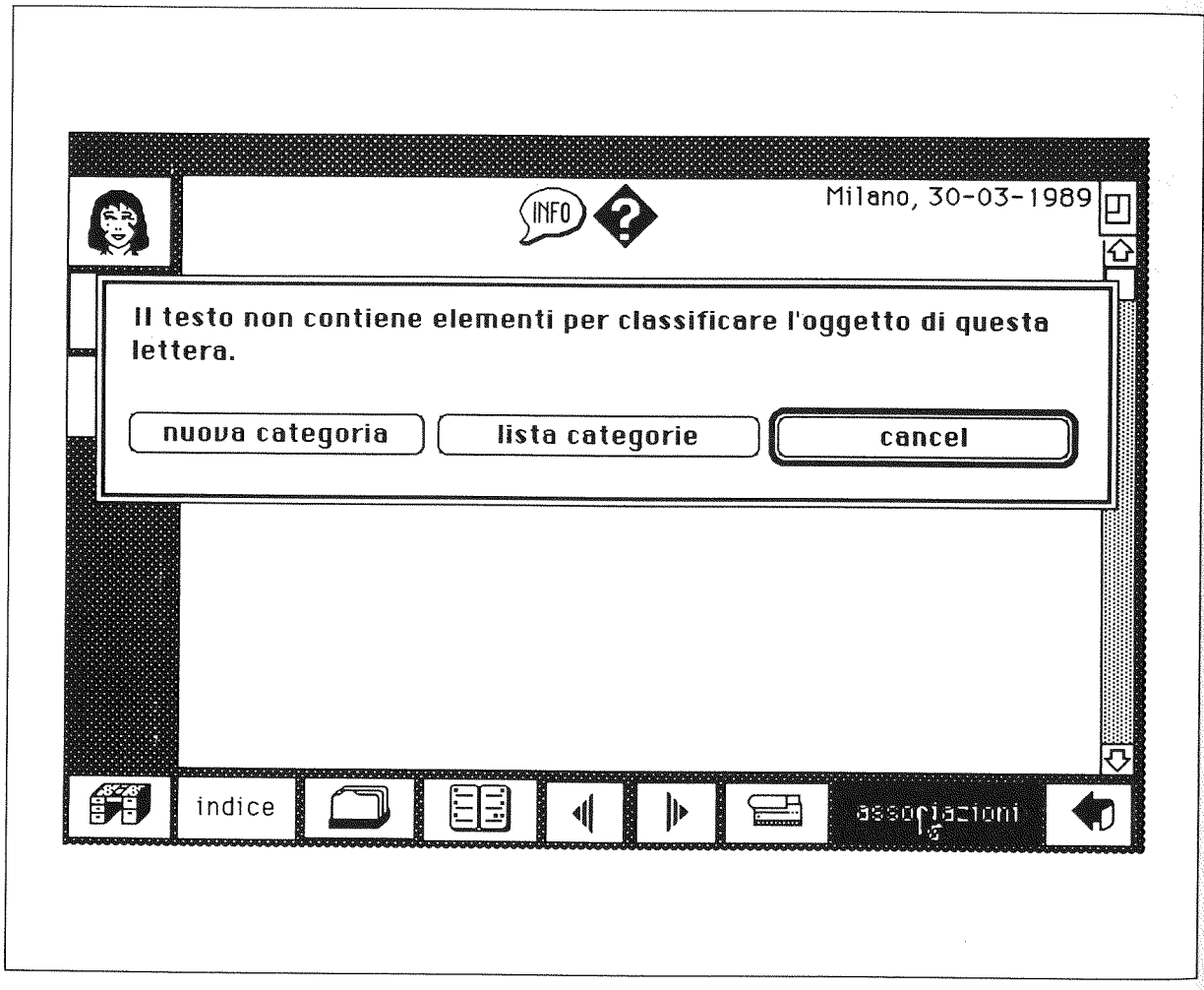


Figure 6

sferring or displaying one or more letters with common elements.
The desk's drawers can be taken also as navigation buttons, because they allow the user to go into the hypertext, defined by system nodes and links; therefore the user is able to move from one environment to another, following the way determined by links and nodes, defined by the scripts and going through it in any direction, activating specific buttons, and so allowing a backtracking of activities.

The drawer's choice defines the archive, whose index is displayed. From within the ***Index Environment*** it is possible to display, upon criteria selected by the user, information regarding the letters. It is also possible to perform some operations, such as:
- selecting a document
- modifying a document
- transferring a document.

Incoming mail makes the incoming-mail's drawer active, and then it becomes open. By clicking on the icon drawer, the user can obtain the display of the arrived documents' list.
Many other objects lie on the desk, such as:
- the ***Agenda***, an object that creates a link between the actual working environment and the ***Agenda Environment***. This environment, dedicated to the annotations of meetings and expiring dates, is represented like an agenda of the real world.

Entering Agenda, coming from different job environ-

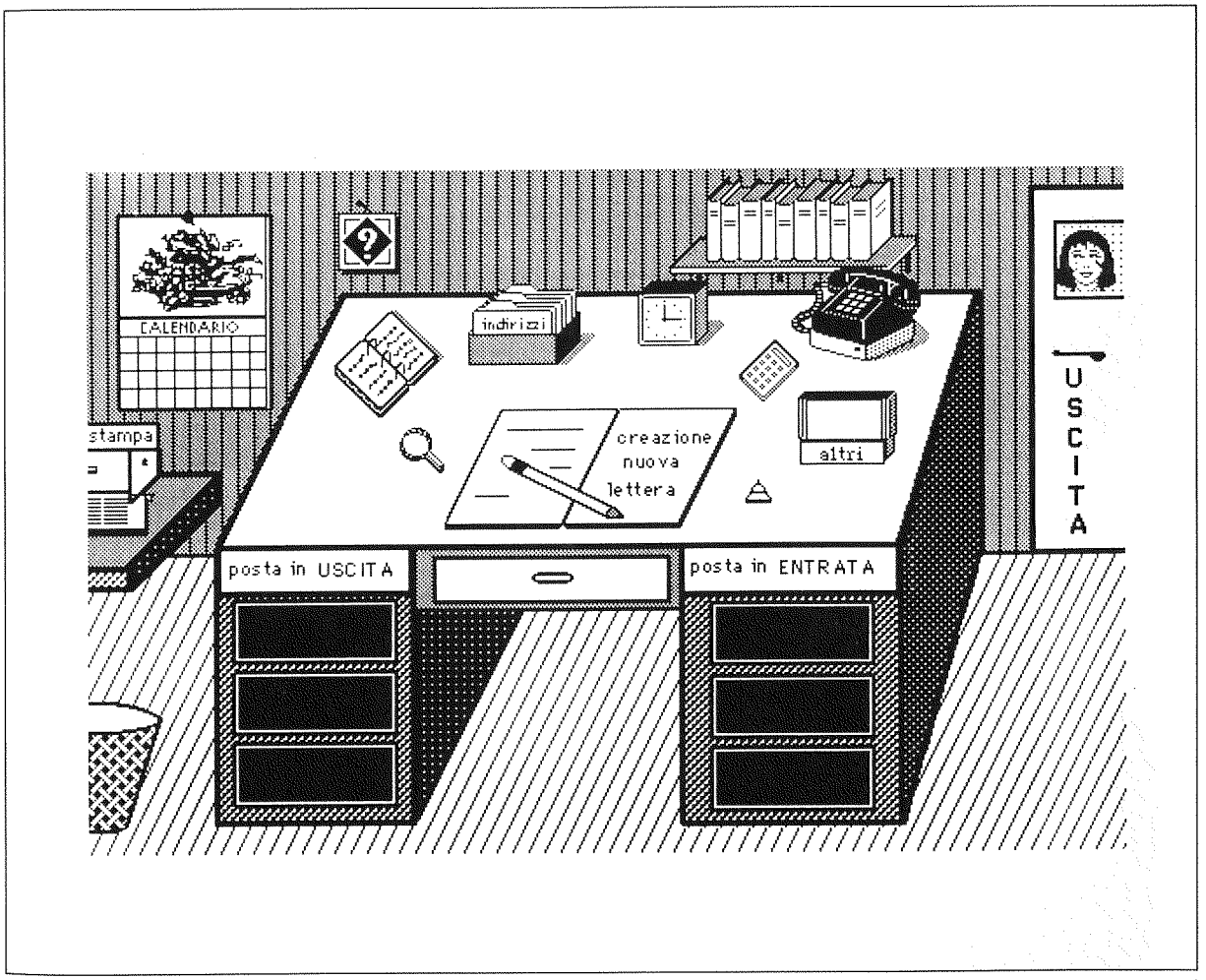


Figure 7: Desk-drawer

ments, can be done either by the user or by a particular system agent (the Secretary Agent). Then information can be inserted, appointments can be checked, notes can be added or modified.
A good example of this procedure is offered by the recognition of a meaningful date during the analysis of a document's text. Due to this the Secretary Agent informs the user about the possibility of inserting text parts, referring to that specific date or any other parts which have something to do with it. On the user's acceptance, this operation is automatically done by the Secretary Agent.
Even in this environment it is possible to perform retrieval activities upon receiving information in different fields. As a result, a weekly page carrying the retrieved information is displayed on the screen.
The ***address archive***: when activated, it requests the user to insert the name that has to be found and retrieved. If the search is successful then the card bearing all the information is displayed. Otherwise, the user is informed and advised to add the name in the address archive. This new name will automatically be inserted in the newly created card.
A script, performing orthographic and syntactic corrections of the data inserted by the user, is associated to every field.
In this environment it is possible to memorize general personal data; and, throughout the activation of specific

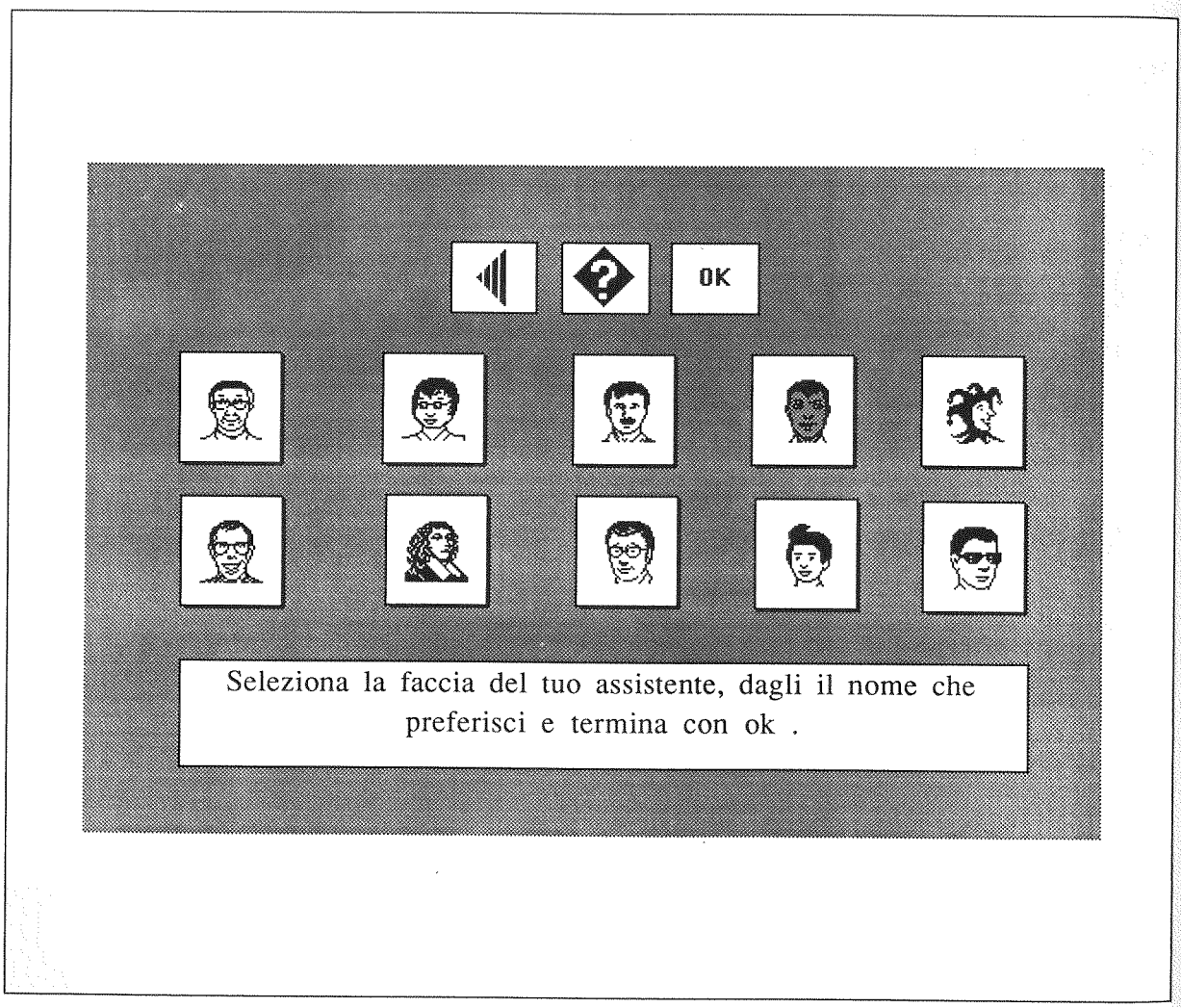


Figure 8: Assistants' definition

icons, additional information about the names can be inserted. This information can later be used by other environments.
The card is formed by these three textual fields:
- name
- address
- telephone number

and by buttons providing different functions.

Selecting the telephone number field, the execution of a telephone call is made possible.
The buttons, represented by arrows or by one alphabetic letter, allow the navigation into the stack.
There is also the magnifying lens that, as previously said, allows a retrieval activity, even on the fields hidden to user.
On the desk there is another class of buttons, which are operative buttons. They are:
- the ***telephone***: this button, represented by the telephone icon, performs the following tasks:

- retrieves the telephone number of the name inserted by the user from the address archive. If this information is lacking, the system informs the user and allows the insertion in the telephone Archive without changing either environment or the context;

- dials the telephone number, executing the call.

Block-notes and pencil is only one object that allows

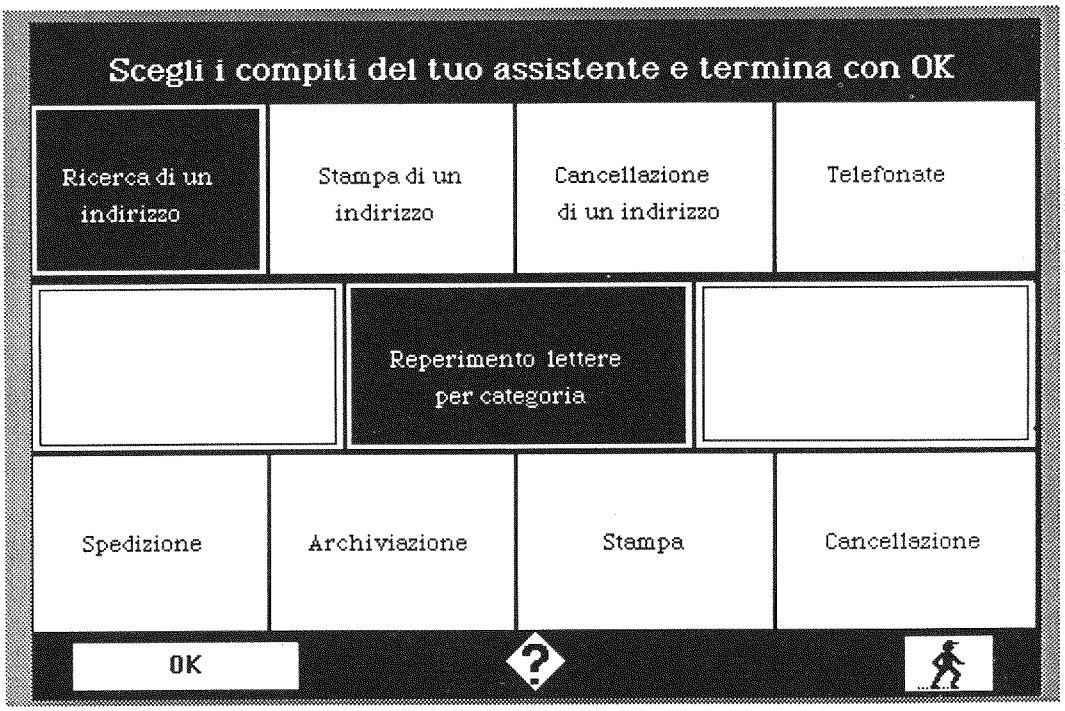


Figure 9: Assistants' tasks definition

the user to enter the document's creation environment. We will discuss this further on.
The ***Secretary agent***: this object represents a button classifiable in different ways: this one, in fact, depends on the context and it is simultaneously an operative and a signal button. The functions performed by this agent in the Desk environment are different from those performed in other environments:

- telephone call booking
- alarms booking
- assistants' activities

The ***Assistants***: the buttons representing the collaborator agents defined by the user are displayed in two different situations:

- activating the ***Interphone*** button
- asking the Secretary Agent to show the list of assistants and their jobs.

The operations done by assistants are defined by the user in the environment dedicated to the composition of assistants' jobs. Activating the agents through Interphone, the operations defined in the associated script are carried out, one by one.
In the desk environment there are some other buttons, defined as signal-buttons because they inform the user about the system change status as the result a particular event. Often the user does not see these buttons until the system, upon recognition of an event that can be important for the user's activity, displays them (e.g.: ***the geographical*** globe and ***the envelope***).

5. THE SECRETARY AGENT AND ITS ACTIVATING EVENT

In the previous chapter we described a first brief analysis of functions provided by this system. Now we will introduce a new concept, the agent intended as alter-ego of the user's collaborators [4].
We can easily imagine a manager who, arriving in the office to do his normal job, would like to know what happened during his absence.
For this the secretary's assistance is necessary.
Therefore the system proposes the collaboration of the Secretary Agent, who "shows up at the door", communicating to the user the information useful for his job.
The Secretary Agent can be activated by :
1) the need to inform the user about events which have taken place, Here are some examples :

- incoming (arrived) mail: if the user so chooses, a list of information relative to incoming letters is displayed in a window. Activating with the mouse a selected document's heading , the document's complete text will be displayed.

- meetings or activities of the day.

At any time the user can enter the Agenda environment but, it is the secretary's task to inform him about appointments. In this way no useless movements will be made between environments. Therefore the Secretary Agent proposes, in a window, the view of an agenda's page only if there are appointments during the day. Activating with mouse the specific signal arrows, the user will be able to see the activities to be made, without changing the environment.

- the update to be made on the current archive: the secretary agent, after checking the existence in the current archive of documents created or modified dating back more than two months, proposes to transfer them to old-document archive, so that retrieval activities will be easier. The user, after having examined the texts, can select letters that have to remain, for different reasons, in the current archive. The secretary agent will go back to her tasks, at the end of this session.

2) by user's explicit request,through hand-bell activation.

The ***hand-bell*** is present not only in the Desk Environment, but also in the document's creation one, allowing the user to call the secretary and proposing the list of those activities which she is able to perform in this specific context.
The scripts associated to the Secretary button can be defined as "elementary agents", because they perform, upon the system requests, only one activity, depending on the context. Hence it follows that the secretary is not an elementary agent, because she performs many activities. Rather she may be considered a supervisor and a coordinator of one or more elementary agents.

6. OTHER COLLABORATOR AGENTS

Describing the secretary agent, a new metaphor of the real world's assistant has been defined. The agents created in the present system, as we will describe later, are able to perform more than one function defined by the user. The newly created agent, a collaborator's alter ego, represents a versatile way to improve the system's working capability in terms of office activities.
The user, in the Desk Environment, can call his collaborators through the two following ways :
- directly, clicking on the interphone icon, placed on the desk. On the interphone's screen the face of the first collaborator and a strip are displayed. By clicking on him both his name and his readiness to carry out the jobs he is able to perform will be displayed. Upon the user's confirmation, he starts his job.
- indirectly, through the Secretary Agent's help. However, in any case, the agents's capabilities can always be displayed in a summary card.

7. DOCUMENT CREATION ENVIRONMENT

It is possible to enter this environment from the desk by activating the block-notes button or by means of a letter existing in the current or in the old-document archive. The letter's environment presents a central portion, like a sheet, for the document composition, and a part in which the different buttons' icons available to the user are represented. Among them, we can find many buttons already seen in the Desk Environment, that here perform other activities, because the context in which they work now is different : hand-bell button and delivery button, not yet examined.
For a better understanding of the functions of the different buttons, we will discuss further the letter and its significant elements.

7.1 The letter

For making the documents recording-and-retrieving easy, it is necessary to identify every document by means of a numeric code and some significant elements.
Taking into consideration that the system has to take care of the handling of literal documents, following the Empiric Research done at the Department [4], we have first of all considered the documents' logical structure, generally recognizable as follows:

- Heading
- Text
- Greetings sentences

In this structure, research has allowed us to identify the document's significant elements. Among them, for the carrying out of this project, the following have been considered primary :

- creation date
- address
- subject
- expiration
- priority

Every new letter, created through the block-notes activation, is presented with the creation date already inserted, modifiable by the user. This date is automatically associated to the document by the system, as a classification element. Other significant elements, such as address or subject, are inserted by the user during the letter editing.

The attributes are recognized by the system upon the activation of specific procedures of analysis. Obviously, the same can be activated also when the user enters the letter from an existing document, retrieved in one of the archives. He can make any modifications and, after confirming to the system the new document's creation, he files this one while the original document text will be preserved unchanged in the archive in which it has been found. The letter analysis can also be made upon the user's request, during the editing or at the end of a document, calling the secretary by means of the hand-bell.

We have just examined some activities performed by the secretary agent in the desk environment and we have explained that these activities are different depending on the environment, because different are the contexts in which she is called to work. Therefore in the letter environment, besides the functions for booking alarms or telephone calls, she proposes :

- information retrieval
- letters retrieval
- analysis

By activating the last function, those elements on which an analysis can be performed will be listed in a window. Once the choice is made, one element is selected and the agent whose specific role is to perform on that type of analysis is called.
It will look into the text for those elements which are meaningful to the letter classification and to its relation to other documents having common attributes. The types of analysis offered by the system are the following:

- receiver analysis
In order to recognize in a letter the name or the forename of receiver, a search for key-words for identifying the requested string takes place. The system provides a dictionary containing some of the possible strings, such as the title that can precede the receiver's name. The analysis is then performed by examining any word one by one and checking if it is in that dictionary. The following string is taken as the receiver's name and is compared with the list of those known to the system receivers. During this operation an orthographic check is also performed to verify if name and forename inserted in the letter are correct: if the system finds an error, even if it is a partial correspondence with an existing string in such a list, then it proposes to modify it. Upon confirmation by the user this correction will be performed every time the error is found. After having retrieved in the receivers' list the one corresponding to the analyzed letter, the system looks for its address into the ***Address Environment***. The same will then be inserted into the letter, always upon the user's confirmation. An address modification performed by user causes the address archive to be automatically updated. Similarly, new names or addresses can be inserted into the address list or into the address archive, respectively.

- priority analysis
The system looks into the text for all occurrences of words whose root is "urgent" in order to classify the analyzed letter under a letters' category having "urgent priority". If no one word containing this root is found, then the letter is automatically classified into the "normal priority" letters' category.

- subject analysis
During this kind of analysis, the system looks in the text for a string allowing it to classify the letter on the basis of a specific subject.

- expiration analysis
A date can be looked at as an expiring event to be inserted into the agenda or as a container (a name) of all the events related to it. The analysis made by the system first considers the date as an information to be inserted into the agenda. As in the case of receiver analyses, there is a dictionary of stored strings that are compared with this information and eventually an association is made.
In this way it is possible to recognize an expiring event even if it is not a numeric date: for example, the expression "at the end of the month" is automatically associated with the last day of the current month. Upon a successful comparison, the secretary agent proposes to the user the insertion into the agenda of a note referring to that date.

- letter analysis
This function allows us to analyze all information which is in a letter, following modalities previously described. The elements recognized in a text by means of analysis functions, allows us to classify, to record and later to retrieve documents much more carefully than we could by using only the creation date or protocol number assigned by system.

7.2 Categories

Every previously described element corresponds to a category to which all letters having defined element are associated.
Beyond elements identified through document analysis, the system offers the possibility of creating new categories that may identify further classification elements of a letter.
This function is present either in the letter environment, which is activated by clicking on the associations button as well as in the index, activated by the categories button. In both ways the activation produces a system card visualization on which those categories eventually previously by the user are duly displayed in a window. Here below is an example:

Organization (meeting, seminary, conference)
where the string preceeding brackets is a categories name while words in brackets form further specifica-

tion in order to recognize, within a letter, those keys that allow the user to classify the letter under a specific category.
The category's name itself is a classification element.
In the same window the one into which the categories list is displayed there is the "new category" button, which - if activated - allows the user to insert the name that is to be assigned to the new category. After the definition of a new category's name, the system proposes to the user the optional insertion of one or more specification.
They can be defined by:

- characters string (key-words) that will be looked at in a letter text
- by the negation of a category's name previously defined
- by (), whenever some specifications are not yet defined, but may be inserted later.

When the letter's analysis is performed, the system verifies whether a significant element of the letter is present as a definition in any category. This is done by comparing not only the names of categories, but also the specifications of the category, provided that the same are not preceded by the logic operator "NOT". As a matter of fact, these kinds of specifications are classified differently.
If we supposed, for example, that a category "CAT" has been defined, and item1 plus item2 specifications are associated to it, we can then make a new category, "CAT2", defined as CAT negation:

CAT2 (not CAT) ====== CAT2 (not item1, not item2)

To this new category (CAT2) all those letters into which, during analysis no specification belonging to the other category (CAT) has been identified will be associated.
However, it is always possible, in the ***Documents Creation Environment***, to associate a letter to a specific category, even when the analysis has not identified any correspondence with the name or with specifications. The user can make associations. Afterwards, the Secretary Agent has notified him about the absence of significant elements in the letter. In this case the system proposes in the window the selection of the category to which the user may like to associate that letter. At this time it is also possible to create a new category without changing the context.
Moreover, the user can modify the name or the specifications of the category: this operation may cause the loss of the relation that was assocating one or more letters to such category. Therefore, in this case, the Secretary Agent informs the user and, upon his request, again performs the analysis of letters associated to the modified category. This activity is carried out also when the user decides to delete the name of specifications of a particular category.

7.3 Delivery

A letter can be forwarded to one or more addresses. In the environment representing undelivered mail, there is always a button, represented by a mailbox icon, that allows the user to choose whether or not to send the letter to one or more addresses.
In the letter case the user is asked to choose in "Receivers" the names to which the letters are to be sent: the system then provides the printing of the document after this insertion.
Letter-sending does not incude its filing. In fact, the user may like to save only the letter's paper-copy. For this reason the system, after sending, asks the uesr to confirm or not the letter filing.

7.4 Filing

The system offers two functions for letter filing, both of them depending on the context in which the user is developing his activity:

- after letter sending: through activation of "Delivery button", the Secretary Agent proposes to file the sent letter; in this way the user can file only the letters important for his activity.

- before sending: through activation of the "Associations button", the letter delivery is proposed to the user. The letter filing causes the updating of the letter's informative record, which - if no analysis has been performed - retains only the system automatically inserted values. The script related to "associations" button performs a letter analysis, inserting in the record all values necessary to define the letter associations. Du-

ring this analysis the Secretary Agent takes care of informing the user of the values which have not been identified in the letter so that they will not be considered as document classification elements.

8. ARCHIVING FILES AND RETRIEVING TECHNIQUES

During the editing of letters it is sometimes useful to enter an archive of previously created documents, one which can be used as a model for composing the letter. To solve the problems of managing many documents the following three archives have been created:

- Mail Undelivered archive, containing letters or messages not yet sent or those that need further changes.

- Current archive, storing letters or messages which are relevant for user's activity.

- Historical archive, in which letters or messages preceding certain date are filed (or that are not important for the user)

The user, by activating the graphic representation of desk drawers, causes the execution of two different functions:
the retrieval of one or more documents, or their transfer from one archive to another.
The letter's significant elements and related information memorized on record, allow the user to find the letters in the archives under different criteria.
The retrieval of one or more letters may occur from more than one working-environments, by means of different techniques depending on the context:

- from the Desk Environment: here as we know the "magnifying lens button" which allows the user to find letters specified by the user is always present. The key-word retrieval begins from Undelivered Mail in the current archives because they contain the most significant letters for user's activity. If no retrieval is performed in those archives, the Secretary asks the user to confirm whether or not to continue the retrieval in the historical archive: this kind of retrieval will request more time because this archive is very big. When the letter is retrieved a new icon representing a closed envelope is displayed. This envelope, following the desk objects management methodology of the system, can be moved with the mouse, on the screen, that have some areas which can be "recognized" by the envelope itself.
When it is moved to a screen area that is not recognized by the envelope, it moves again to its original position. Otherwise, some procedures are activated so it is possible to delete, to print or to find the retrieved letter simply by moving the envelope button onto the relevant icon, such as waste-basket, printer or desk-drawer.
Clicking on the envelope icon makes it possible, as in the real world, open the envelope and visualize of the whole document in a specific window.
Now the user can directly enter the letter or, at will, he may continue the retrieval without selecting again the key-word but simply by acting a button script.

- from address card-index: activating one of the three buttons represented by sending mail's desk drawer icons, the corresponding index is displayed : this one is formed by the list of letters and related information.

In this context it is possible to visualize the letter under selective criteria. For example, selecting a receiver's name, the list of letters sent to the same address, and related information will be displayed. Obviously, this will happen for every field of the index structure.
By activating the arrow icon, placed at the left of the letter information, it is possible to directly enter that letter.
The document retrieval, under selective criteria, is useful not only for this opportunity (i.e. entering a letter, without browsing the whole archive), but also for some other different functions offered to the user.

In fact, from the index environment it is possible to select more letters and then to execute their deletion, printing or transferring (procedures).

- letter creation environment: in this one, letter retrieval is one of the tasks performed by the secretary agent. By activating the hand-bell, a list of functions that the secretary is able to performs "letters retrieval" included, is displayed. The retrieval, as we have already seen, is executed under the letters' classification criteria : this means that, for example, on selecting a retrieving

under "priority" criteria, the retrieval of the letters bearing a priority equal to the one in the creating letter will be proposed. Of course, the user may insert another kind of priority, under which it is possible to retrieve the associated letters.
The list of documents found is displayed, also forming a link to all other letters which are in the same list. Now the user can, using the mouse, select a letter.
That letter, to avoid the context change, is displayed on the same Hypercard card from which the retrieval has been carried out. In this way the user can copy parts of the selected letter and insert them in the letter that is being created. Clicking twice on the line of a chosen letter causes its activation. Thus a relation to the one which the user is working on is made.
The possibility of creating links between letters allows the connection with hypertextual methodology of documents stored in the three archives. So the user, during the editing of his document, can examine letters, without knowing their allocations in the archives.

9. ASSISTANT WORKS DEFINITION

After having discused agents as user collaborators, we talked about the user's possibility to create new collaborators, and specifically to instruct them about particular more or less complex task . Here below we will see how this is made possible. Activating the specific button, the user can enter the environment into which he can define a collaborator agent, satisfying his needs in a particular context. Every system's agent is universally defined by:
- a name: defined by the user, to identify it and recall it from other environments
- a face icon: chosen by the user, to define it and recall it from different environment.
In this way the system allows the user to personalize his collaborators, avoiding in every kind of environment in which he is working any ambiguity about the choice of assistant to be recalled. The collaborator's definition environment is divided into two subenvironments. In the first one the user can select a sequence of elementary jobs, among those displayed, and assign it to his assistant. The proposed jobs will help the user in two different job's environments:

- address card index: in this environment the assistant can make telephone-calls, retrieve names or addresses, upon use of a specific key-word chosen by the user and then, a user's will, print and/or delete them. If the user by mistake selects the deletion as first activity, the following choices will be disabled (because it is not possible to perform activities on deleted data).

- in the letter environment: in this environment too the first task that the agent performs is a retrieval activity. The retrieval can be carried out on different criteria :

- key-word
- category
- archive

By selecting one of those entries, the following possible tasks will be displayed :

- sending
- filing
- printing
- deleting

Like in the addresses index-card environment, here too there are some rules which disable wrong operation sequences :

- delivery disables printing, because it is included in a selected activity

- deletion disables all other activities, provided it has been selected as the first item

- filing disable delivery, because it is a part of an activity performed by itself

After the selection of activities that are to be assigned to collaborators, it is possible by clicking on the "OK" button to enter the second subenvironment.
Here the user can choose the face icon that will represent his assistant and, later, insert the name assigned to him in a specific field.
Upon conclusion of these operations, the user, pressing the "OK" button, completes the personal assistance definition.

At this point the desk environment is shown, and the user can recall his assistant for giving him confirmation

about the carrying out of the jobs for which it has been defined.
Every one of these assistance-agents is nothing but an assembly of procedures that the user can combine, using also logical operators, as long as he likes. Substantially it is a new way of user-programming. Therefore, even an unexperienced user will be able to create suitable functions for different situations according to his needs.

Acknowledgments

The author would like to thank Prof. G. Degli Antoni for his helpful suggestions and encouragement. Many thanks are due to Roberta Macchi and Hug (Hypertext User Group) for supplied documentation.

10. REFERENCES

[1] G. Caja, P. Piardi, C. Testero, S2: UNA SEGRETERIA INTELLIGENTE PER LA GESTIONE DI DOCUMENTI DI UFFICIO, Convegno A.I.C.A. Text Processing IV - 12/13 Dic. 1989

[2] G. Degli Antoni, Le Van Huu, LA AUTOMAZIONE D'UFFICIO COME MECCANIZZAZIONE DEL RAPPORTO TRA TEMA E DISCIPLINA, Convegno AICA su "Modelli di Ufficio",1982

[3] G. Degli Antoni INTEGRAZIONE DI NORME E PROCEDURE NELLA VITA D'UFFICIO, Convegno FAST su "Integrazione di Informatica e Diritto", 1983

[4] P. Piardi, UNA INDAGINE EMPIRICA SU UN CAMPIONE DI CORRISPONDENZA, Note di Software n.32, 1986

[5] F. Laminafra "SISTEMA S: IMPLEMENTAZIONE DI UN SISTEMA DI SEGRETERIA ELETTRONICA", Tesi di Laurea in Scienze dell'Informazione - A.A. 86/87

[6] A. Forti "Sistema S: progettazione di un sistema di segreteria elettronica", Tesi di Laurea in scienze dell'Informazione - A.A. 86/87

[7] G. Degli Antoni, IL COMPUTER, IL REALE E L'ARTIFICIALE, Note di Software n.41, 1988

[8] A. Boder, HYPERCARD: AN EXAMPLE OF HYPERTEXT COGNITIVE ASPECTS, 1988

[9] G. Degli Antoni ARTIFICIAL WORLDS, Atti Convegno HUG - Secondo Incontro,1988

[10] G. Caja, P. Piardi, C. Testero RASSEGNA SUGLI IPERTESTI, Note di Software n.39/40,1988

[11] C. Testero, UN SISTEMA COGNITIVO PER LA AUTOMAZIONE D'UFFICIO CHE REALIZZA UNA NUOVA METAFORA D'AGENTE - PARTE PRIMA, Tesi di Laurea in scienze dell'Informazione - A.A. 88/89

[12] G. Caja, UN SISTEMA COGNITIVO PER LA AUTOMAZIONE D'UFFICIO CHE REALIZZA UNA NUOVA METAFORA D'AGENTE - PARTE SECONDA, Tesi di Laurea in scienze dell'Informazione - A.A. 88/89

VISUAL-SYMBOLIC INTEGRATION IN NEURORADIOLOGY (*)

Francesco Bonadonna
Cattedra di Fisica Medica e Informatica Medica
Facoltà di Medicina
Università degli Studi
Palermo

RIASSUNTO

La radiologia una disciplina in cui l'integrazione delle immagini e della loro descrizione simbolica assume un'importanza fondamentale. In questo lavoro viene analizzata la struttura logica dei database per immagini esistenti nei moderni sistemi di diagnostica per immagini, con particolare riferimento ai tomografi a risonanza magnetica (MR).
Viene quindi proposto un nuovo inquadramento teorico, proveniente in parte dalla struttura dei database per immagini e in parte dai sistemi ipermediali.
Infine viene presentato un prototipo funzionante, realizzato in linguaggio Pascal su una personal workstation, e vengono discusse le sue e i possibili contesti di utilizzazione.

ABSTRACT

Radiology is a discipline in which the integration of images and their symbolic description reaches a fundamental importance.
In this paper the image database structures as existing in the modern imaging devices are analyzed, with particular reference to the MR tomographs.
Then a new theoretical framework is proposed coming partly from the image databases structures and partly from the hypermedia systems.
A working implementation of this theory, implemented in the Pascal programming language on a personal workstation, is presented and its performance and utilization scenarios are discussed.

1. INTRODUCTION

Radiology is characterized by a peculiar integration between visual knowledge, i.e. images, and symbolic knowledge, i.e. image descriptions, diagnosis and related information. Computer science offers useful tools for dealing with the specificity of the diagnostic imaging language. In the simplest case this corresponds to an electronic version of conventional approaches, resulting in CD-ROM based electronic textbooks or PACS. An evolution toward modalities impossible to reach by conventional media comes with the hypermedia, whose fundamental innovation, for what concerns the visual-symbolic integration, is the possibility to create, inside the images, active regions to be used for establishing associative links to other images or texts.
This paper is centered on the generalization of this concept. A first step implies the integration of complex symbolic structures to the image regions, such as descriptions with several fields, logical links and also programs.

(*) This work was the subjet of a degre thesis "Un sistema ipermediale per la gestione della conoscenza visiva in Neuroradiologia" by Giovanni Fulantelli. The thesis was discussed at the Dipartimento di Scienze della Informazione of Milano on the 30th april 1990. The advisors were G. Degli Antoni and F. Bonadonna.

Further steps, now at an experimental stage, imply a 3D extension of the visual-symbolic integration, characterized by a smaller granularity than the active region method. Moreover, the labelling techniques, necessary to instantiate the visual-symbolic integration structures, such as the semiautomatic edge detection by threshold or gradient method, evolve from simple image processing techniques to more sophisticated image understanding ones or to the self-organizing approach of neural networks.
Finally, the software and hardware factors that limit the practical implementation of these techniques are discussed, and a possible utilization scenario is outlined.

It's very important to note that the system proposed relies on the interaction with the user as a fundamental resource. In other words, it could be considered as enhancer of human thinking and not as emulator. This is also referred to as **"man-machine joint reasoning"** and is similar to a typical team work situation in which a set of software tools perform as specialized assistants that work together with the head of the team.

2. LOGICAL STRUCTURE OF VISUAL AND SYMBOLIC DATA IN COMMERCIAL IMAGING DEVICES

The logical structure of images and symbolic (textual) data in the commercial imaging devices is very important to know because it is something the radiologist is used to seeing and hence it's a convenient starting point for any new organization. The "conceptual consistence" with the imaging devices (such as MR and CT tomographs) and also the textbooks that are commonly used by the radiologist is a key feature for a system that has to be of practical use. Obviously the conceptual consistence exists just for common concepts and ends where new structures begin.
As sketched in figure 1, the organization of images in a commercial tomograph is typically a hierarchic one. The whole database contains a set of "examinations", each corresponding to a complete patient study performed in a single session. Patient data such as age, weight, height etc., belong to this level and are contained in specific database fields.

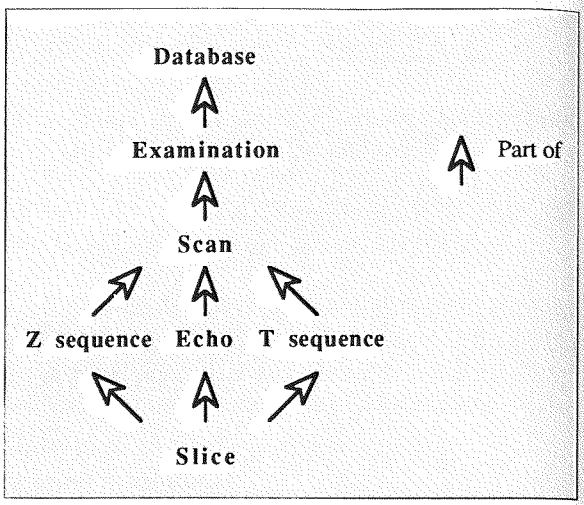


Figure 1: The MR tomograph database hierarchy. Examination is a complete patient study. Scan is a particular acquisition, with a specific set of parameters. Echo is specific for NMR and consists of one set of images for each echo. T (temporal) sequence is the result of dynamic studies. Z sequence is a sequence of 2D images along the third spatial dimension (Z axis). Slice is a single 2D image

One examination includes a set of "scans" each corresponding to a particular acquisition identified by a unique set of parameters like anatomical orientation (Sagittal, Transferral, Coronal), tilt angle, resolution, stimulation (SE, IR etc).
A scan may include one or more "echo sequences' that are a particularity of MR acquisitions and consist of different images obtained in the same region and at the same time but with different "echo times".
A temporal sequence is the result of a dynamic acquisition. A Z sequence includes one or more bidimensional images (slices) in spatial sequence along the third (Z) axis.

It's important to note that the information contained in fields (at the patient, examination and scan level) could be considered as symbolic "knowledge" for differentiation with the visual knowledge represented by the images, but it should be better indicated as symbolic "data", because a more complex organization is required to justify the transformation from data to knowledge.
A typical example of such complex structure is the

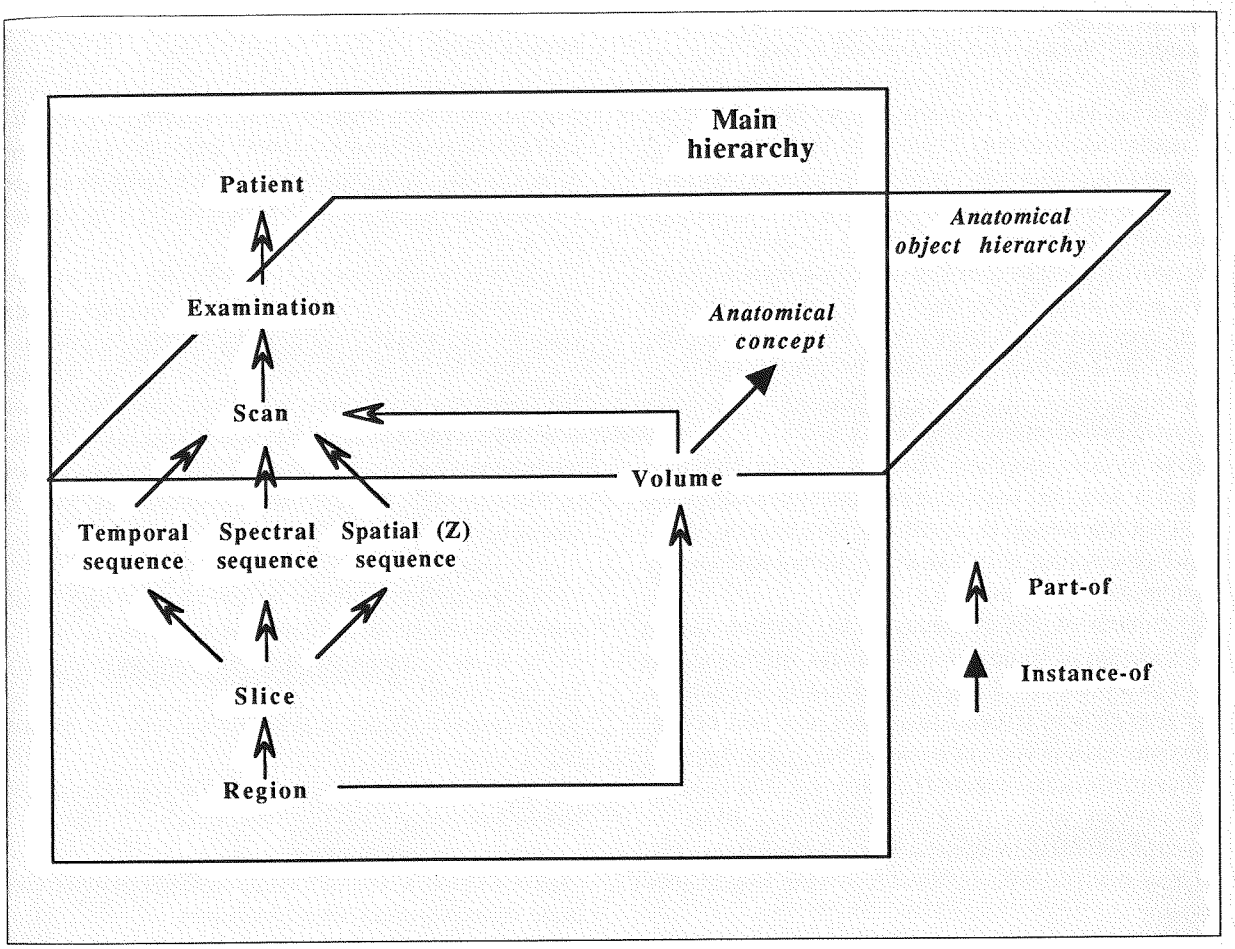


Figure 2: The main "examination oriented" hierarchy. Patient contains all the examination of a patient. Examination contains a complete patient study. Scan contains a particular acquisition, with a specific set of parameters. Spectral sequence corresponds to an "echo" in MR, hard/soft rays or before/after contrast media in Xray CT, to different radiotracers in Nuclear medicine etc. A group of spectra is also called a "spectral sequence" Temporal sequence is a slices or a group of slices acquired in the same time. Spatial Z sequence is a series of slices ordered along the Z axis. Region is a part of a slice and is a complex structure containing graphic information, database fields and also executable code (activities). Volume represents the intersection of the main database hierarchy with the anatomical one

network of logical links, typical of a knowledge base.

Images as a whole in the MRI databases: Images are the bottom of the hierarchy in figure 1 because there are no parts of an image that could be addressed individually in the database. Of course it's possible to draw regions of interest (ROIs) into the images, but there is no possibility of including them into the database as specific objects with their own names and to recall them.

One of the main reasons for developing a new image database structure is the extension of the hierarchy to the image regions identified by ROIs. The presence of this new level will allow database searches based on region specific data such as area, mean density, spatial position, shape etc.

3. VISUAL-SYMBOLIC INTEGRATION

In figure 2 a hierarchy is represented that includes

regions on the images and new structures such as "patient", "region" and "anatomical concept".
The "region" is the key feature of this system because represents the link between image areas and their symbolic description [1,2]. Regions, manually or semi-automatically drawn on the image, are generally corresponding to an anatomical structure or to a specific alteration or sign in the contest of a structure.
Patient is a level that contains all the examinations made by a given patient. This level is not necessary in small databases that cover just a few days of work like that included into the MR tomograph, but is useful for large databases where is possible to find more than one examination for the same patient.
In the system proposed, there is another hierarchy together with the "scan centered" one: the **anatomical concept hierarchy**. It reproduces the common anatomical description, with classes for normal or abnormal anatomical objects etc (fig 3).
The bottom of this hierarchy is the volume i.e. an instantiation of an anatomical concept belonging to a particular scan. A volume is composed by one or more 2D regions which, extending what was said before, represent the connection between the image database structure (the scan-centered hierarchy) and the conceptual network whose presence transforms a large but unstructured set of image and region data to a knowledge base that could be explored following logical relations.

4. ACCESS TO EACH HIERARCHY LEVEL: WINDOWS

Ergonomy is very important for a system that overcomes the experimental phase and has to be used in the routine work. An application of ergonomic concepts is the use of specific windows for exploring each hierarchy level.
Figure 4 shows the directory window that allows user interaction with the scan-centered hierarchy. After opening this window, the radiologist sees just the patient list. He selects one patient from the list (names have been reduced to initials for the figure) and the list of all the examinations for that patient appears. Then the user chooses one examination and the correspondent set of scans appears. Finally a scan is selected and several things happens:

1)The spectra list (echoes for MR images) appears if more than one spectrum has been acquired.
2)The temporal sequence appears if a dynamic study has been done
3)The spatial Z sequence appears if more than one slice has been recorded.
4)The list of volumes defined for that scan appears (if almost one volume has been defined).

Figure 4 shows a case of a multi-echo acquisition with three slices, repeated after 15 minutes

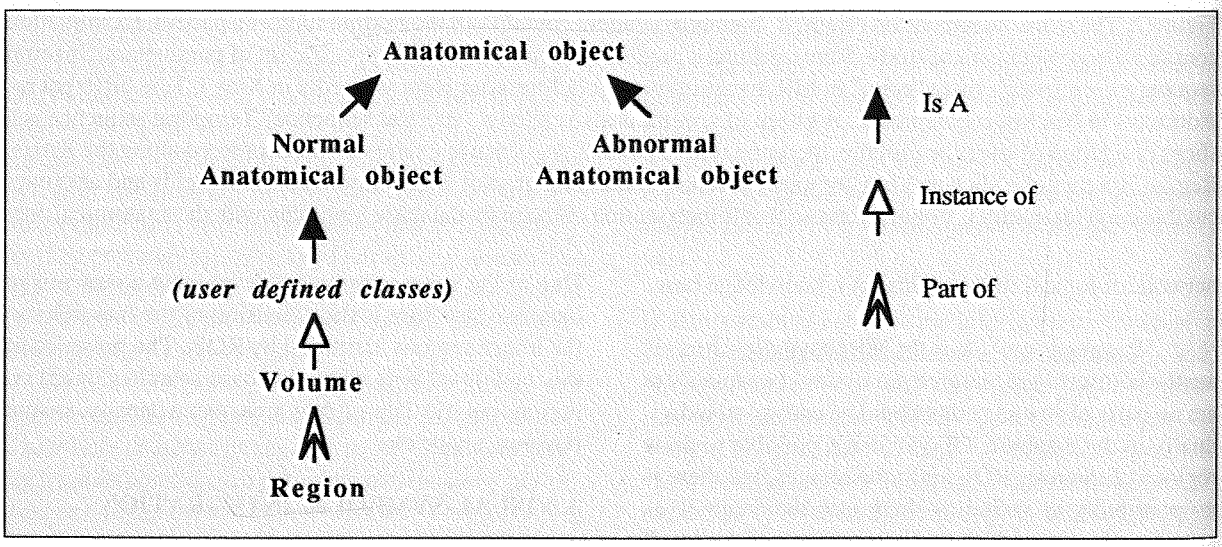


Figure 3: The anatomical objects hierarchy

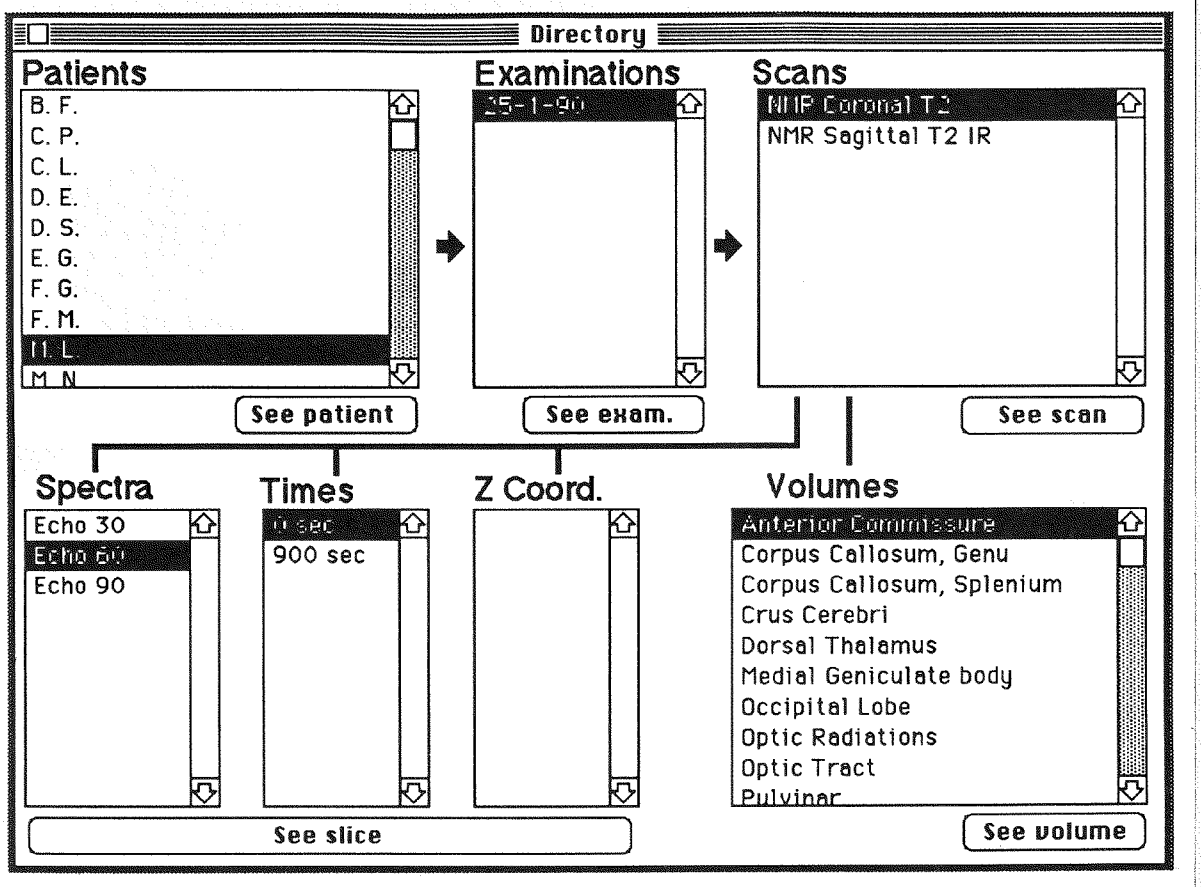


Figure 4: The directory window used for exploring the scan-centered hierarchy

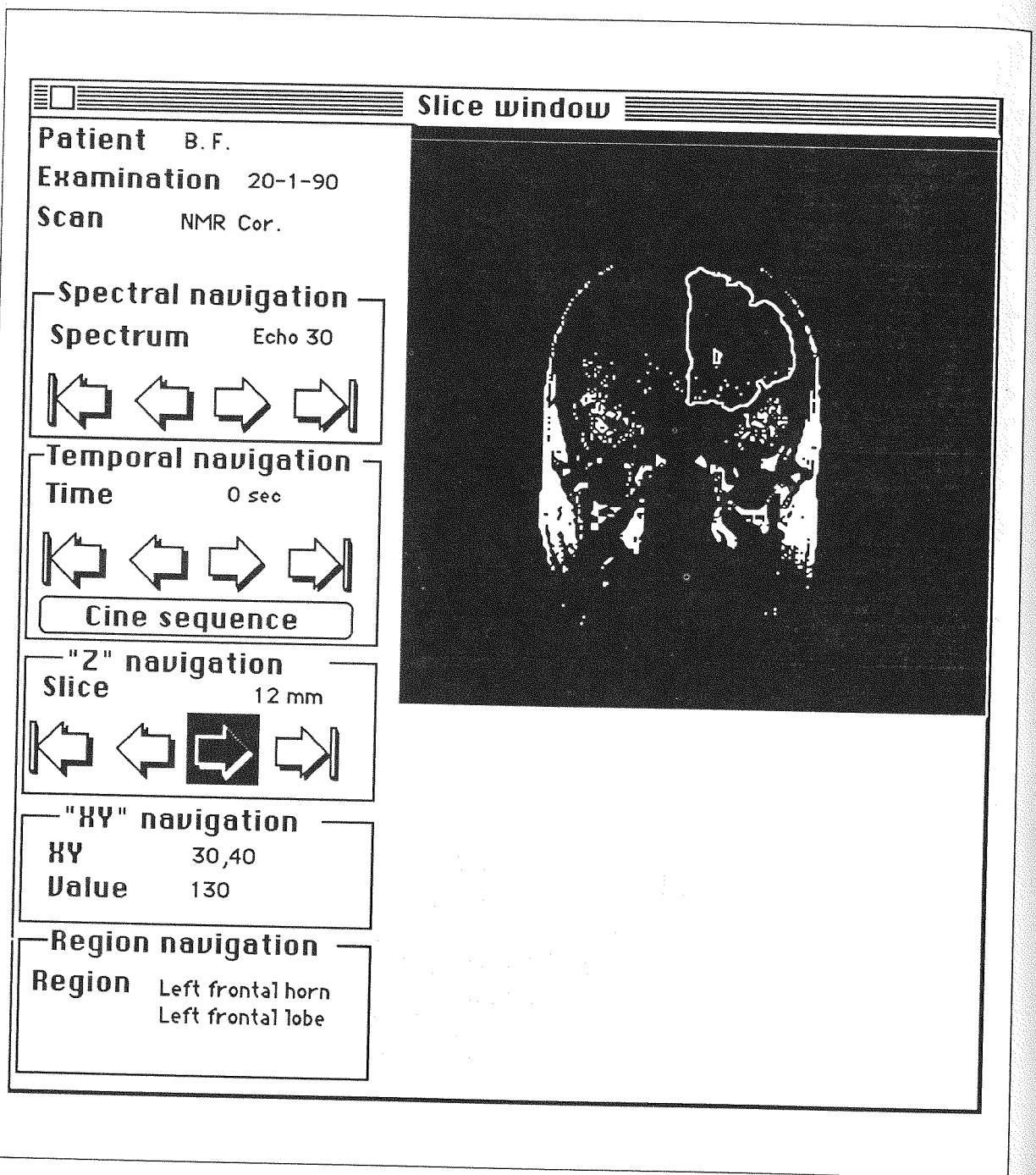


Figure 5: The slice window. The areas on the left correspond to the different possibilities of navigation in in spectra, times and Z coordinates. XY navigation gives the cursor coordinates over the image and "Region navigation" gives the name of the region (or regions) touched by the cursor. In this figure The cursor is over the frontal horn of the left lateral ventricle, and is inside a wider area corresponding to the left frontal lobe of the brain. Both the frontal horn contour and the left frontal lobe contour are enhanced

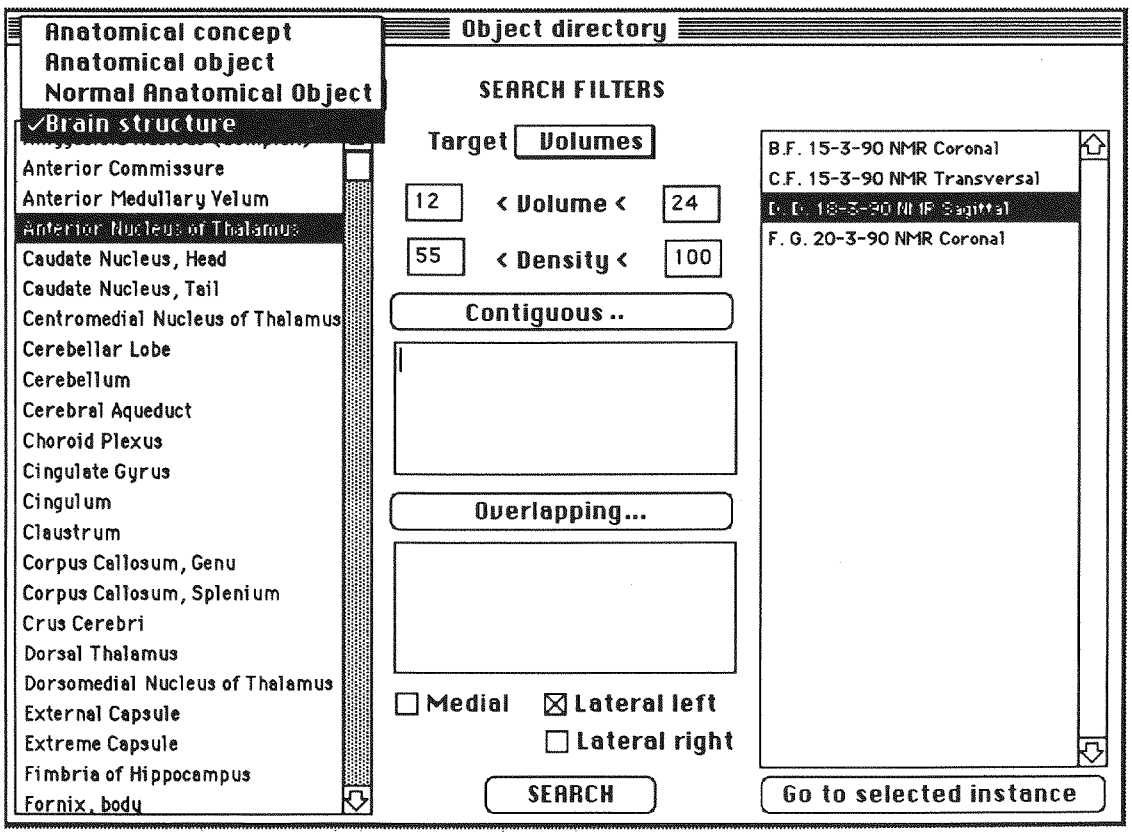


Figure 6: The object search window. On the left there is a hierarchical list of anatomical concepts. The user can select one of them to search for all its instances that appear in the right list. In the middle are the tools used for setting conditions on the content of the database fields of regions or volumes

By default, the first of each spectrum, time and Z coordinates is selected and the user can go directly to that slice using the "see slice" button. The user can also choice another slice by clicking on the appropriate items of the list, as shown in the figure 5.
The slice appears in the "slice window" (fig 5).

In the slice window several navigation tools are present for exploring the image itself and the other in the same scan:

1) Spectral navigation. The four arrows brings to (in order) the first, the previous, the next and the last of the spectra (echoes for MR images) for the same spatial and temporal position. In practice, if the user starts from the slice at 12 mm at time zero, in echo 30, he or she will obtain all the other echoes at 12 mm and time zero.
2)Temporal navigation. Uses the same logic of the spectral and Z navigation. A special command "cine sequence" shows sequentially all the images belonging to the same spectral and spatial position .
3)Z navigation. The arrows drive the navigation in the spatial series, maintaining the same spectrum and time. Note that slices may be contiguous or separated by a gap that varies from 10% of the slice thickness to several times the slice thickness. Obviously Z sequences with gaps bigger than about 50 % of slice thickness couldn't be considered as continuous sequences, also if this fact affects mostly the 3D reconstruction rather than the direct observation.
4)XY navigation. This and the region navigation have no commands because the actuator of the navigation is the pointer driven by the mouse. In XY navigation are reported the 2D spatial coordinates of the pointer over the image and the corresponding pixel value.
5)Region navigation. This is an exploration of the symbolic description of a particular pixel. If the pixel corresponding to the pointer belongs to a region defined on the slice, the region perimeter is enhanced and its name is displayed in the "region navigation area". If the user clicks the mouse button when a region is selected, then a special "region window" for more complex interaction appears.

5. REGIONS AND VOLUMES

As shown in figure 2, a region is a part of a 3D structure called volume. A volume is *"instance of"* an anatomical concept. The relationship instance of links an abstract concept to its particular realization: i.e. the concept "left frontal horn" to the left frontal horn of the patient BF, seen in the coronal scan in date 20 Jan 90. The presence of this link is very important because it allows the particular instance to inherit all the knowledges belonging to its class.

5.1 Object oriented database search

Database fields: Together with links, regions and volumes also have database fields for recording their specific parameters. In this implementation, mean density and area are the fields for the regions, and mean density and volume are the fields for the volumes. Fields are used in database searches aimed to identify the regions or volumes matching a given set of constraints.
Hierarchical list: The anatomical object hierarchy may be used for constraining the search to a given class of objects or just to find all the instances of a concept. Interaction with the anatomical object hierarchy is based on the hierarchical list that is shown on the left of figure 6. On the top of the scrolling list there is a pop up menu that shows a path through the IS-A hierarchy. When an item, corresponding to a class, is selected all the components of that class are displayed in the list. For example, in figure 6 the class "brain structure", subclass of normal anatomical object, is selected and all the brain structures defined into the system are shown in the list.

Regional procedures: a third type of search on the image content is based on region specific data that are not recorded in the database fields but calculated during the search itself. The areas "overlapping" and "contiguous" in figure 6 are examples of this kind of data. Because regions may be defined in each moment and in each order, the list of the structures that are contiguous to a given one is changing, so it is calculated just when needed.

6. UTILIZATION SCENARIOS

The system may be used in different ways:

•As a **database** it complements those present in radiological dedicated workstations. The possibilities of

defining and search for regions give a lot of new possibilities. Moreover the performance/price ratio of magnetic and optical disks is the most convenient due to the large diffusion of personal computers.

•As a **didactical tool** it behaves like an active radiological textbook in which the regions shows their name and are enhanced when touched. It's also possible to search for the images in a given pathology, for the pathologies in a given region and much more using the object database tools.

•As a **diagnostic support system**, it could be used in a way similar to that described for didactical use. Moreover it's possible to search for a lesion which has similar features to that found in the patient. Is important to stress that the routine diagnostic work, with the little overload of creating and labeling regions around pathological or interesting objects, may become a continuous building of a radiological image knowledge base [3].

7. TECHNICAL NOTES

This program has been written by Giovanni Fulantelli in object Pascal for the Apple Macintosh IIci personal workstation with 25 megahertz clock, 8 megabytes RAM and 80 megabytes hard disk. The images come from a Microvax II which is part of the Philips Gyroscan T5 Magnetic Resonance tomograph. The personal computer is linked to the Microvax via an RS 232 cable at 19200 baud and emulates a terminal VT100.

8. REFERENCES

[1]Bonadonna F., De Maria M., De Simone GF., Mercadante G., Pappalardo MP., INTEGRAZIONE DELLE CONOSCENZE VISIVA E SIMBOLICA IN NEURORADIOLOGIA, Neurologia 1989, Atti VIII Congresso Nazionale dell'Associazione Italiana di Neuroradiologia, a cura di Kurt Pardatscher, Edizioni del Centauro, Udine, pp 377-38231

[2]De Simone GF., Bonadonna F., Barberi G., De Maria M., Mercadante G., Pappalardo MP., Rabiolo A., DATABASE PER IMMAGINI E STRUMENTI INFERENZIALI PER LA NEURO-TC: TEORIA ED IMPLEMENTAZIONE, Neuroradiologia Oggi. F. Smaltino, R. Elefante, S. Cirillo (ed), Idelson, Napoli, 1988, pp 315-320

[3]Greenes R. A., KNOWLEDGE MANAGEMENT AS AN AID TO MEDICAL DECISION MAKING AND EDUCATION: THE EXPLORER—I SYSTEM, Proc. MEDINFO 86, Elsevier, Amsterdam, The Netherlands, 1986, pp 895—899

RESEARCH NOTES AND COMMUNICATIONS

DATAGLOVE AND AUDIO VISUAL ART

Adriano Abbado
Independent Artist
Milano

RIASSUNTO

Fra le tecnologie emergenti che si offrono ad un artista, il DataGlove e' certamente una delle piu' eccitanti. Sono quindi descritti gli strumenti utilizzati per realizzare una performance di immagini e suoni controllati da un DataGlove. Sono infine trattati ulteriori miglioramenti, in parte realizzati, dell'idea originale.

ABSTRACT

Among the emerging technologies that are offered to an artist, the DataGlove is certainly one of the most exciting. The tools to create a performance of images and sounds controlled by a DataGlove are described. Further improvements, in part realized, of the original idea are also exposed.

1. INTRODUCTION

Among the emerging technologies that are offered to an artist, the DataGlove is certainly one of the most exciting. At the meeting "Nuove Immagini e Formazione", held at Venice's Palazzo Fortuny at the beginning of December 1989, I had the opportunity to create a short performance of images and sounds controlled by a DataGlove. This has been possible thanks to the Milan-based company ARS, that gave me the possibility to use a DataGlove. The Venetian experience has led me to further improve the original project. This paper shortly describes my work with the DataGlove and the consequent perspectives.

2. THE TOOLS

In order to create my performance in Venice, I interfaced my software "AV", that I had written before, with a DataGlove. The software "AV", written in LightSpeed C, runs on a Macintosh II, and permits the creation of 2-D abstract objects, that can be either animated or static. It is divided in two major parts: the editor and the player.

The editor allows the user to create an object at the time. An object is made of a shape filled with a colored pattern. The pattern, in turn, is made of n polygons. Each polygon can be of different color, and can be drawn with the 8 pen modes allowed by ColorQuickDraw. In the case of animated objects, both the pattern (i.e., each polygon of it) and the containing shape are created by interpolating a starting and an ending frame. Afterwards, a trajectory can be defined, linear, curved, or both combined. The object is animated on a background that can be either a flat color, usually black, or

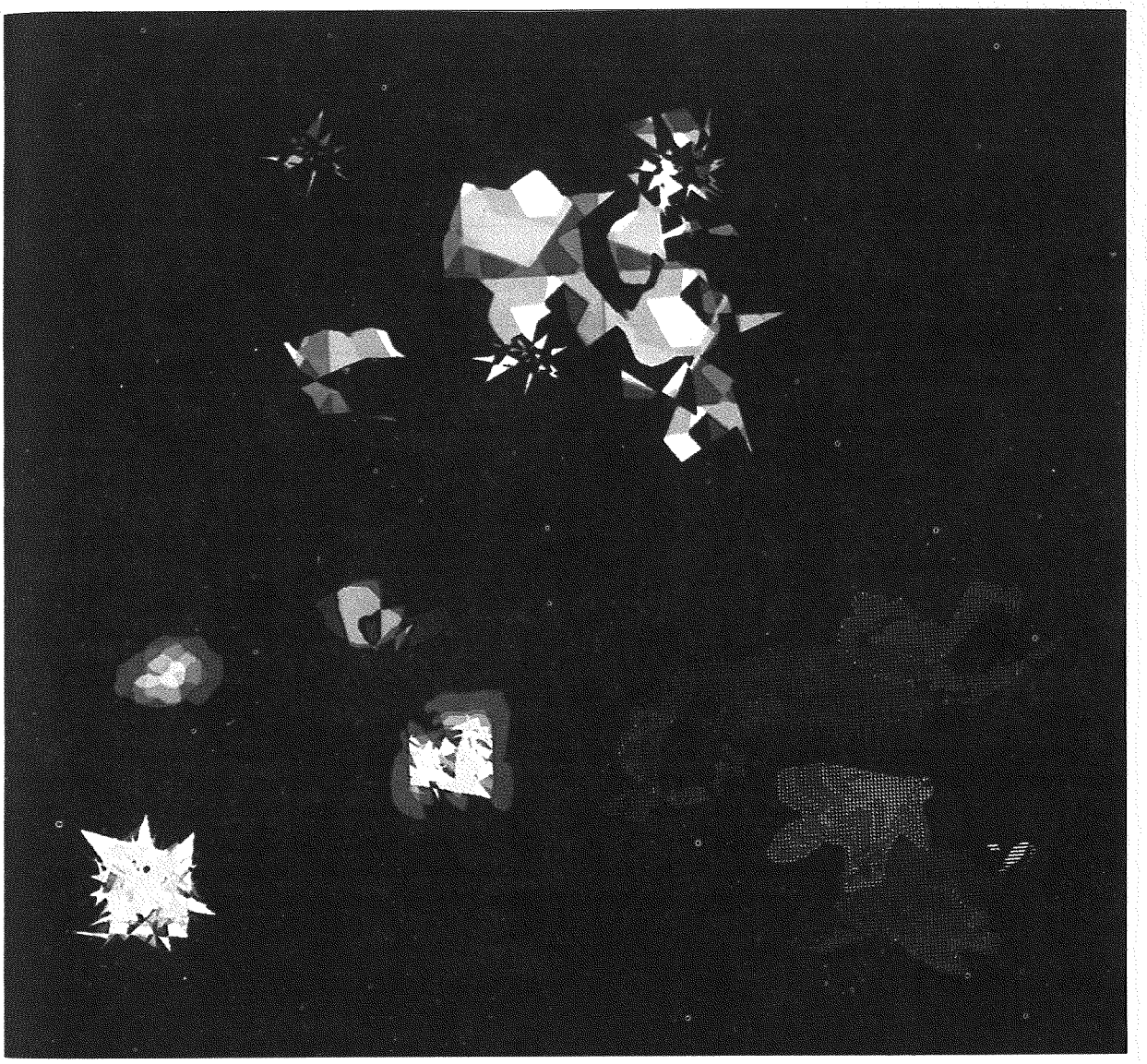


Figure 1, 2, 3

a PICT image imported through the clipboard (fig. 1, 2, 3).
A fundamental feature of this program is that it can play a MIDI musical instrument through a MIDI interface. MIDI, as known, is a standard protocol for the communication between computers and digital musical instruments. Because of this feature, the objects are usually modelled on an already existing synthetic sound, that is then retrieved again, in synch with its object, in the player section.

The player section in fact, as the name suggests, does nothing but assemble the different objects and sounds previously created, and play them together. Several objects can be played at the time, their number depending on their size.

The connection between the player section and the DataGlove allowed the interpreter to control a few short audio-visual sequences through finger movements. For example, bending the index triggered a first sequence,

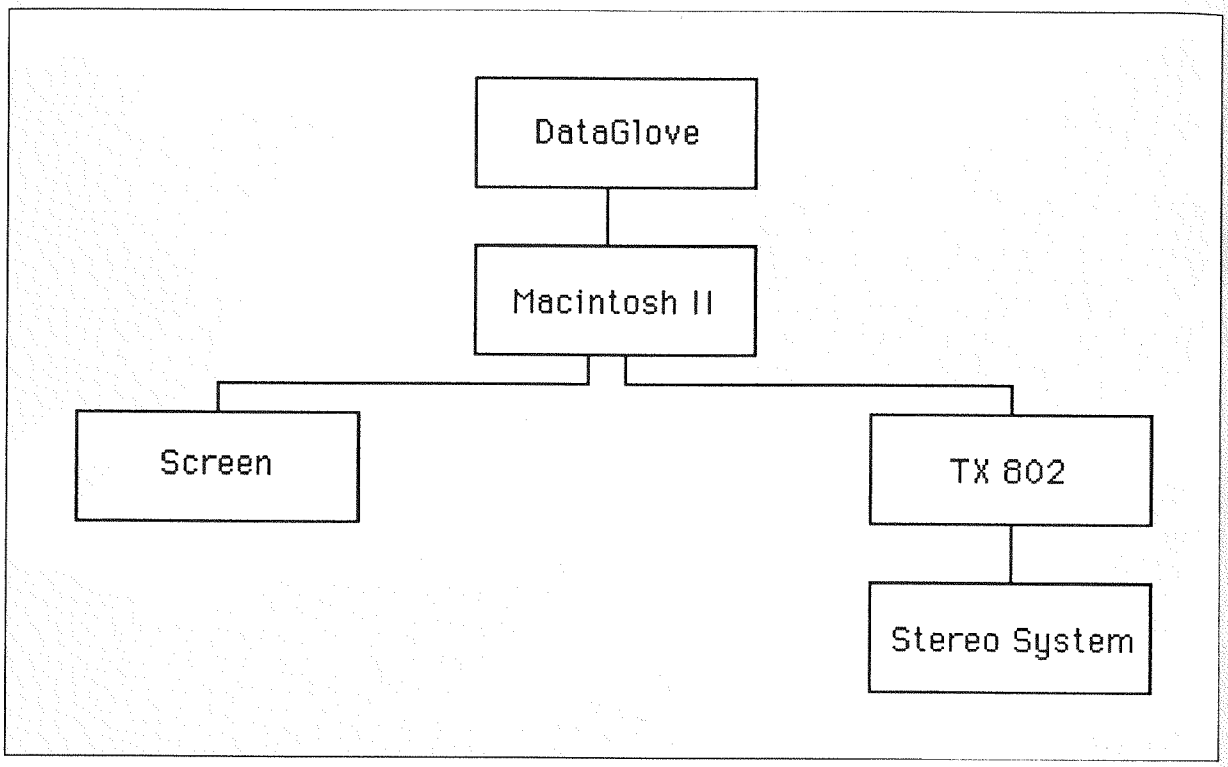


Figure 4

while bending the middle finger triggered a second sequence, and so on, repeating this process at will. The images could be seen on a monitor of a Macintosh IIcx, while the sounds, created on a Yamaha TX802 synthesizer, were heard through a traditional stereo system (fig. 4).

3. DEVELOPMENT

In this sense, the DataGlove has been used only as a switcher.
The performance in Venice, though, was only a first step, the embryo of a more complex idea, now in part completed. In fact, the performance is now organized so that the interpreter, wearing the glove, moves his hand in an audio-visual hyperspace.

The hyperspace can be imagined as a 1 meter wide cube, placed in the real world (i.e., the hand moves in the air). The hyperspace is divided in many tiny cells, each "containing" an audio-visual microsequence (0.5 - 5 sec.), actually stored in RAM. Moreover, cells adjacent on the three axes x, y, z are related one each other. The performer therefore moves his hand and navigates in this artificial world, creating coherent audio-visual events. The images are projected on a big screen by means of a videoprojector.

4. CONCLUSION

The experiences with the DataGlove have been extremely interesting. As always, I have accumulated several kinds of feedbacks, that in turn can lead me to further improvements.
For example, a modification to this project can be represented by the creation of 3-D animation, mainly because we are better accustomed to the ideas of navigation and exploration in a 3-D rather than in a 2-D world. An Rb2 stereoscopic display can also enhance the effect of 3-D images. However, a multitude of Macintoshes would be necessary to reach a relatively huge audience, making the performance a show for few people. On the contrary, if the images are to be video-

projected, the stereoscopic effect would be lost, but a larger number of people could enjoy the performance.

5. AKNOWLEDGMENTS

The perfomance at the meeting "Nuove Immagini e Formazione", was made possible thanks to the company Artificial Reality Systems (Milan), which gave me the possibility of using DataGlove.

6. REFERENCES

[1] David J. Sturman, David Zeltzer, Steve Pieper HANDS-ON INTERACTION WITH VIRTUAL ENVIRONMENTS, UIST '89: ACM SIGGRAPH/ SIGCHI

[2] Apple Computer INSIDE MACINTOSH Volume V, Addison-Wesley Publishing Company, 1988

[3] Wassily Kandinsky CONCERNING THE SPIRITUAL IN ART, Dover Publications, 1977

FROM TEXT TO HYPERTEXT THROUGH METATEXT

A. Limongiello, F. Soresi, A. Spinelli, W. Vannini
Direzione Studi Sistemistici Avanzati
Bull HN Information Systems Italy

The prefix 'hyper' generally refers to 'something having more than three dimensions'. Using this observation as a starting point, we address the issue of *where* and *how* to find these extra dimensions in documents. 'Traditional' documents are usually said to be 'linear' or 'sequential'. This indeed refers more to how *reading* can be applied to such documents than to how they are structured. Anyway, hypertext systems propose to bypass 'linearity' by more or less refined 'linking' capabilities.

The hypertext dimension is added to linear texts by the mechanism of links.

We identify the 'metatext' component in the usual text that gives one more dimension to traditional text. The term metatext wants to address all those components that yield a description of the text like Table of Contents.

With metatext, new dimensions are added to documents through smart use of those traditional tools (see literary devices), whose effectiveness has been perfected througt centuries of use by the scholar community.

In other words, *metatext* are all those literary devices which act as descriptions or references (at a metalevel) of the conceptual structure embedded in the document, such as: notes, references, glossaries, thesauri, tables of contents, analytical indices.

As said above, such literary devices have a deep-rooted tradition. The first well-structured Table of Contents is contained in the first book of the 'Historia Naturalis' by Plinius the Elder. The first known glossaries, on the other hand, date back to the Greek Age, and were used as an aid in the reading of Homeric poems.

In the following paragraphs, we outline how to migrate from traditional 'linear' documents to computer-supported hypertextual ones, through the methodological adoption of metatext.

The traditional division of documents into chapters, sections and paragraphs must be captured by an appropriate formalism, such as (5) or (6).

In our model, margin notes summarise the meaning of concepts presented in the corresponding paragraphs and, if read in sequence, may yield a summary of the whole document (the Table of Concepts).

The role of the Table of Contents is twofold: on one side, it shows pointers into the hierarchical document structure; on the other side, it reveals the conceptual

organisation of the document, as originally intended by the author.

Glossaries, thesauri and analytical indices create networks of text items. Bibliographical references and footnotes are interpreted as standard hypertextual links.

Graphical descriptions of conceptual nets should be expressed using a suitable geometric (object-oriented) graphic language in order to interpret graphic links as hypertext links.

Given a document and its metadescription, the *inherent* network of concepts is easily extracted and mapped onto hypertext structures. This is the basic step to allow sensible migration from textual to hypertextual systems.

Several useful tools can be devised to exploit this extended interpretation of documents: first of all, a Translator able to interpret standard document and metadocument descriptions into hypertext format.

A Conceptual Outline Processor may give a unifying view of different representation schemas. *Syntactical* (chapter, section, paragraph...) as well as *conceptual structures* (conceptual relations) can be browsed (and edited) at different levels of detail, through both textual or graphical representations.

Hypertext is not a tool, but a technology, applied to evolving documentary systems; so we claim that the achievement of a standard for hypertext must arise from: 1) adopting of a standard format for traditional documents as a *lingua franca* among hypertext systems and 2) adopting a precise bijection of current hypertext structures to well-known documentary structures.

BIBLIOGRAPHY

[1] Conklin, J., HYPERTEXT: AN INTRODUCTION AND SURVEY, Computer Magazine, September 1987

[2] Limongiello, A., Soresi, F., Spinelli, A., Vannini, W., FROM TEXT TO HYPERTEXT THROUGH METATEX, in (NIST)

[3] Nelson, T.H., LITERARY MACHINES, Strathclyde 1982

[4] PROCEEDINGS OF THE FIRST HYPERTEXT STANDARDISATION WORKSHOP, Gaithersburg MD, USA, January 16-18, 1990

[5] ISO 8613-1, OFFICE DOCUMENT ARCHITECTURE (ODA) STANDARD, available from International Standards Organizations

[6] ISO 8879 STANDARD GENERALIZED MARKUP LANGUAGE (SGML), 1986, available from ISO

BROWSING IN KNOWLEDGE THROUGH ANIMATION: AN APPLICATION TO THE A.I.D.S. CASE

Mario Di Floriano
Hypertext User Group
Dipartimento di Scienze dell'Informazione
Università degli Studi
Milano

This short note presents a multimedia interactive prototype for accessing and updating a knowledge base related to A.I.D.S. desease.

One of the aims was to analyze the use of interactive animated interfaces for browsing particularly in the case of the A.I.D.S. knowledge.

The work was developed at the Dipartimento di Scienze dell'Informazione (Computer Science Departement) of Milan University jointly with H.U.G. (Hypertext User Group) and I.T.B.A (Istituto Tecnologie Biomediche Avanzate) of C.N.R. (Comitato Nazionale Ricerche).

In a previous work an A.I.D.S. knowledge base system was organized around a Petri net model of some clinical and biological processes related to that desease [2] and [3]. In that system the Petri Net model was a formal guide for accessing informal documentation as well as software tools associated to the net elements (condition and events). Starting from that implementation, the Hypermedia interactive system described here was developed, whereby a continuous animated model has been used as an interactive interface for accessing the formal knowledge (and also the informal one). In this Hypermedia system, the animated model is fully exploitable as a set of sequences of animated moving pictures. The sequences are selected by the user in its exploration of the desease. When the user requires a formal or informal documentation of the process presented by the animation, he stops the animation. As a consequence a Petri Net is shown with tokens indicating the corresponding conditions or events actually occuring in the animated simulation. By clicking the net elements presented, the user can access various documents or notes (even personal ones) associated with the net elements. If nets elements are elements of an high level net, then the user will be able to access to the detailed version of the process (another net) through clicks. In some cases software tools are associated with the net elements, corresponding to states of the animation. Particularly interesting are the information retrieval tools that can be used to haveaccess in a large bibliographical data base (actually the Compact Library: AIDS and the Medline CD-ROMs) the documents that justify or refute the selected formal model. The motivations to the project were:

- the hope to find the way to improve the productivity in research in AIDS through an easier access to the documentation.
- an investigation to a novel approach to knowledge browsing.
- a fairly simple and a systematic new approach to eli-

citation of formal bases and informal documents staring form large data bases.

With this system, previous efforts in information retrieval or modelling are tightly associated to the results of the work (actually a Petri Net). In such a way, updating new documents can be done from the previous model elements to which information retrieval procedures are associated: if the underlying hypothesis will be confirmed modification will not be needed, otherwise modification have to be introduced, according a critical analysis. The adoption of a net representation imply that causal reasoning is used to model the deseases details. So, updating in the formal knowledge base means a modification that the user must introduce in the underlying net and eventually, if needed, in the animated view. In the actual prototype the AIDS-expert user can by itself modify the nets while, if animated sequences have to be modified, some support by an animation creator is probabily required, given the difficulties in designing animated sequences.

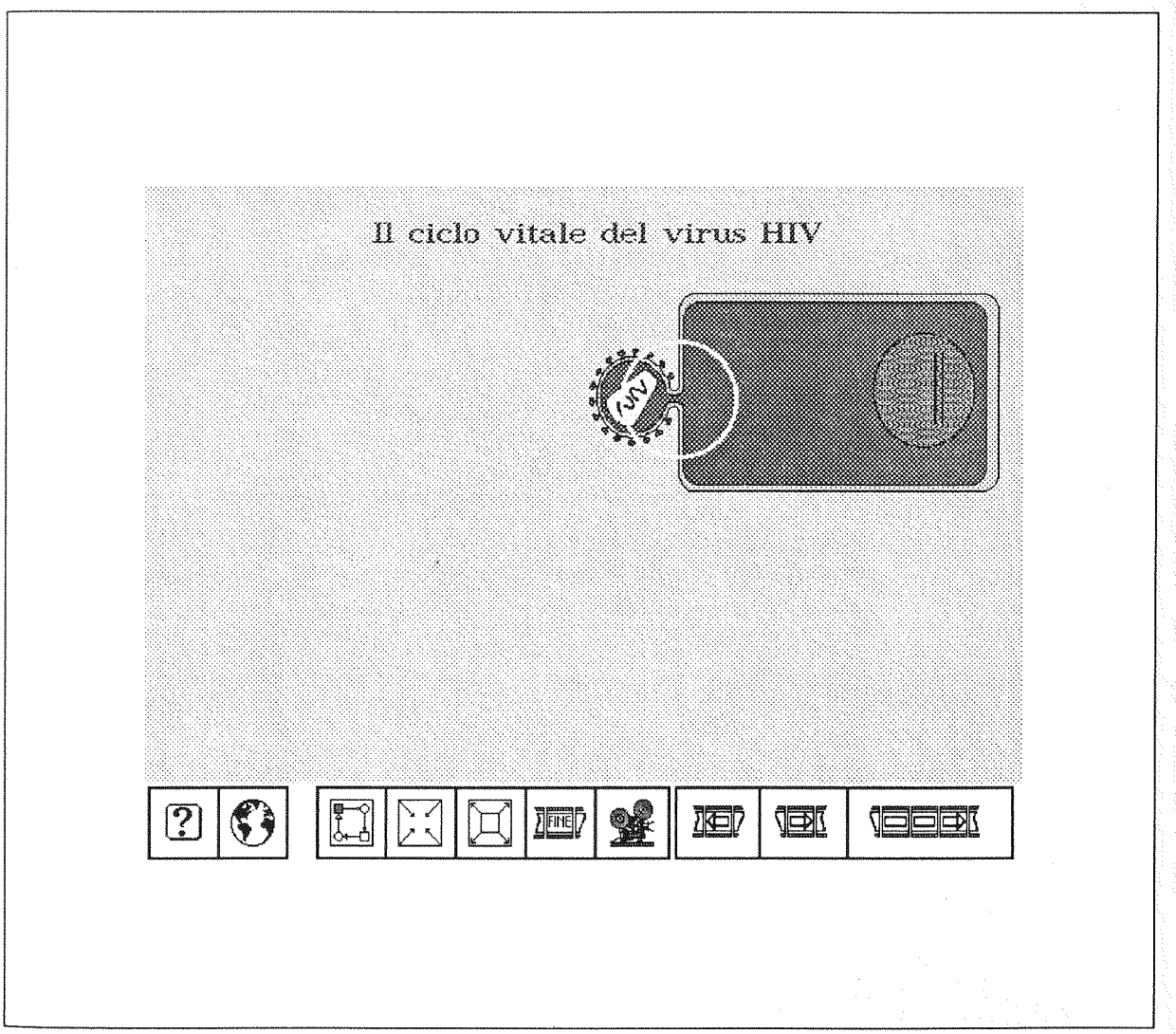


Figure 1: The snapshot corresponding to the animation related to the AIDS virus life cicle. The snapshot shows the situation in which the virus binds on the cell.

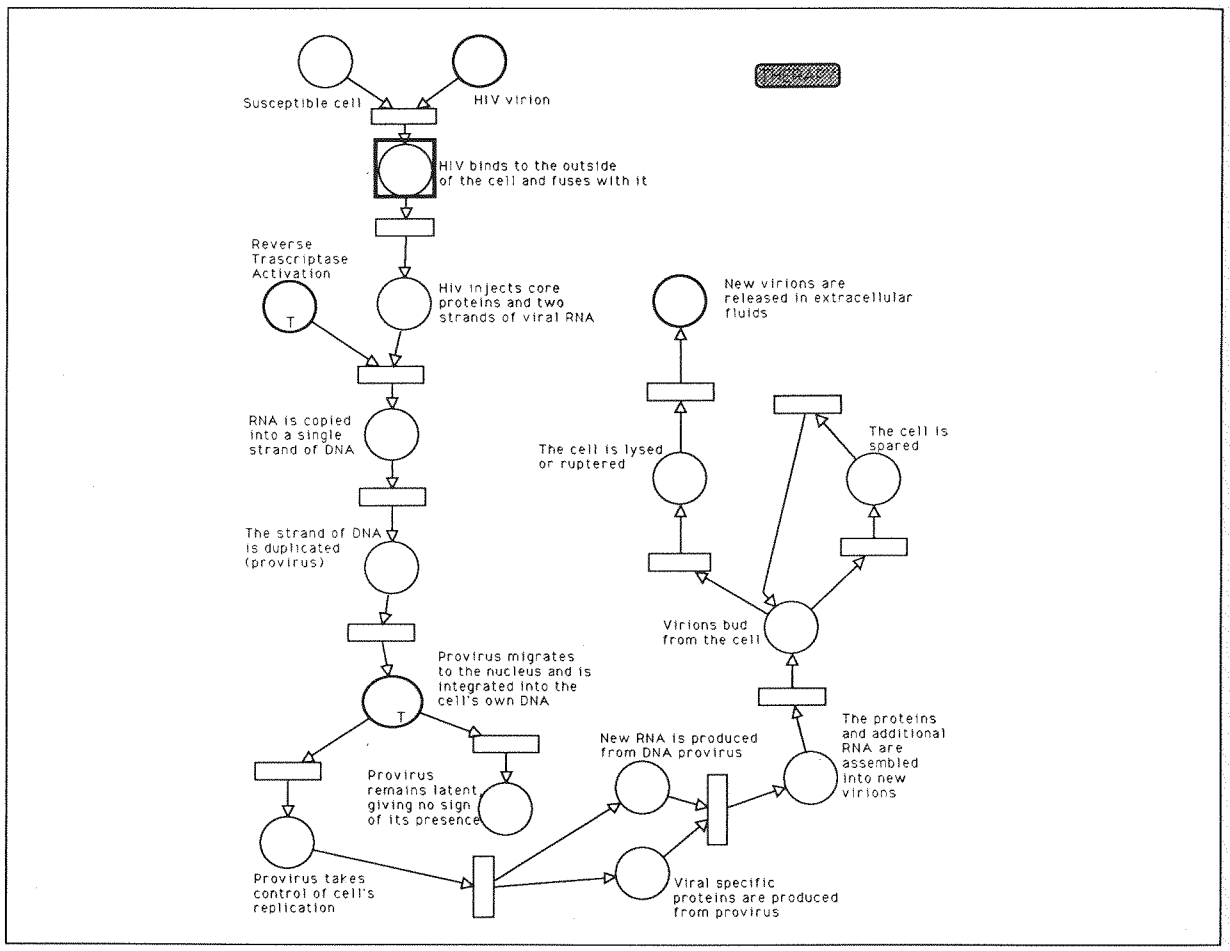


Figure 2: The net related to the virus life cicle. The token indicate the state related to the snapshot of Fig.1

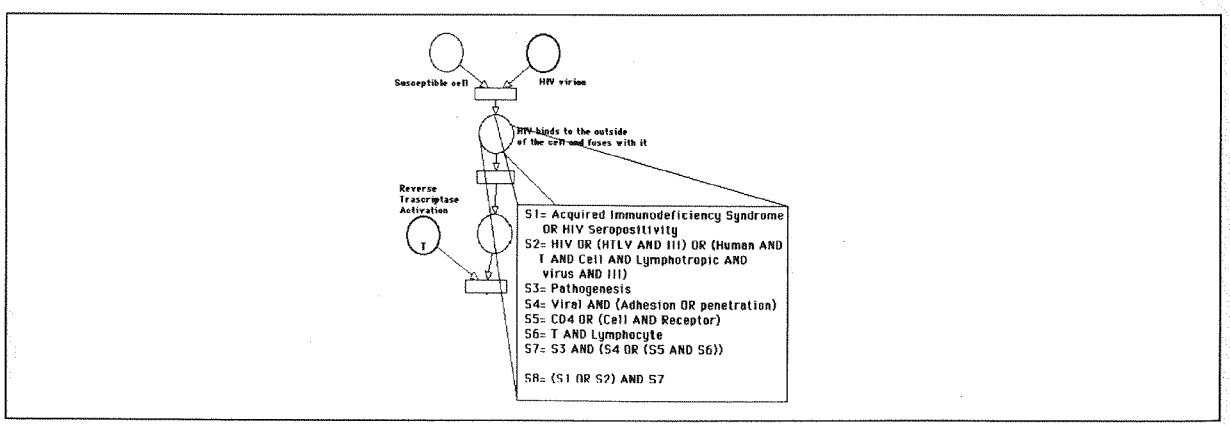


Figure 3: The search path associated with the net state modelling the virus life cicle. This search path selects inside the CDs the documentation about the binds of the HIV virus on a suscettible cell.

ACKNOWLEDGEMENTS

Thanks to Prof. Giovanni Degli Antoni for the basis idea of this project and for his precious suggestions, Prof. Antonio José Rodriguez Selles for his contribution to the analisys and implementation of animation, H.U.G. (Hypertext User Group) and specially to Roberta Macchi for her support and encouraging during these months. I would also like to thank Marco Bondi for the constant collaboration during the develope of the part related to Petri nets, Wilma Barcellini, Monica Attolini and Paola Varesco for the realization of the Petri nets AIDS model and Paolo Valcher for his help during the development of this project.
Finally, thanks to Maria Luisa Villa for her medical support.

REFERENCES

[1] Annals of internal medicine, COMPACT LIBRARY: AIDS: MEDICAL LITERATURE MOVES TOWARDS ITS FUTURE, november 1988

[2] Attolini Monica, Varesco Paola, ORGANIZZAZIONE DELLA CONOSCENZA SULL'A.I.D.S (SINDROME DA IMMUNODEFICIENZA ACQUISITA) IN UN AMBIENTE IPERTESTUALE, Atti del convegno "Verso la comunicazione elettronica", Grande Fiera d'Aprile, Milano 1989

[3] Attolini Monica, Varesco Paola, STUDIO E REALIZZAZIONE DI UNA STAZIONE DI LAVORO INTEGRATA PER LA RICERCA SULL'A.I.D.S., tesi di laurea, Dipartimento Di Scienze dell'Informazione, Milano 1989

[4] Bondi Marco, PROPRIETA' DELLE RETI DI PETRI CON UNO STRUMENTO IPERTESTUALE, Atti del convegno 'Verso la comunicazione elettronica', Grande fiera d'Aprile, Milano 1989

[5] Bondi Marco, LO STUDIO DELLE PROPRIETA DELLE RETI DI PETRI CON UNO STRUMENTO IPERTESTUALE, tesi di laurea, Dipartimento Di Scienze dell'Informazione, Milano 1989

[6] Degli Antoni Giovanni, IL COMPUTER, IL REALE, L'ARTIFICIALE, Note di software N. 40

[7] Degli Antoni Giovanni, ARTIFICIAL WORLDS, secondo incontro HUG, Dicembre 1988

[8] Degli Antoni Giovanni, ORGANIZZAZIONE DELLA CONOSCENZA, Atti del convegno 'Verso la comunicazione elettronica', Grande Fiera d'Aprile, Milano 1989

[9] Degli Antoni Giovanni, FROM REALITY TO HYPERMEDIA THROUGH ARTIFICIAL REALITY, nota interna al Dipartimento di Scienze dell'Informazione

[10] Gallo R.C., IL VIRUS DELL'AIDS, Le scienze 1988, N 223, pag. 31-40

[11] Haseltine W.A., Wong-Staal F., LA BIOLOGIA MOLECOLARE DEL VIRUS DELL'AIDS, Le scienze 1988, N 244, pag. 28-37

[12] Hopgood F.R.A.,COMPUTER ANIMATION USED AS A TOOL IN TEACHING COMPUTER SCIENCE, The Auerbach annual 1975 Best Computer Paper Isaac L. Auerbach Editor, pag. 185-196

[13] Matthews T.J., Bolognesi D.P., VACCINI PER L'AIDS, Le scienze 1988, N 244, pag. 134-142

[14] MacUser Features Staff, MULTIMEDIA, MacUser, March 1989, pag. 88- 137

[15] MacroMind, DIRECTOR STUDIO MANUAL and DIRECTOR OVERVIEW

[16] Ransom Rob, TAKE ACTION, MacUser Europe, November 1989, pag. 36-42

[17] Ransom Rob, ANIMATION IN SCIENZE TEACHING: VIDEOWORKS AND HYPERCARD, Wheels for the mind europe, February 1989, pag. 5-11

[18] Wartow Ronald, INSIGHTS ON HYPERCARD ANIMATION: LIVEN ANY STACK WITH SIMPLE BUT STRIKING GRAPHIC EFFECTS, Macworld, February 1989, pag. 79-92

19) Whitby Max, BRAVE NEW WORLDS, MacUser Europe, June 1989, pag. 28- 35

HYPERMEDIA AND BLIND PEOPLE TODAY

Nunzio La Ferlita
Redco Informatica
Milano

''Multimediality'' is a term which has not yet been well defined in the world of computers. It is agreed that this term signifies a harmonious fusion of Images, Voice and Data; but on how to realize this harmony, which media to use, on the ideal applicative outlets for this tool of the near future, discussions are still continuing.

If the concept of ''Multimediality'' is to be gone into further, we arrive at ''Hypermediality''. In the incandescent magma which pours out of the crater of computer science, one can catch but a glimpse of the glow of the most brilliant solutions, and one can identify the directions in which it is confusingly leading us; but the lucid certainty of a solution or a series of well defined solutions is still lacking.
One thing though seems indisputable: the path of computer science is the same as the path of Man; the overall direction of the magma is towards a future where computers are increasingly integrated and increasing like their makers. Computers, therefore, which manage sounds and voice will be more in evidence and improved; they will accept high resolution images and will be able to process them more and more easily; they will combine increasingly large and better organized quantities of data with sounds and images; and, lastly, they will be increasingly able to transfer sounds, voices, images and data from one computer to another at increasing distances apart and at increasing speed.

But computer science is a product of civilization, and there is no civilization worthy of that name which does not take into consideration its social minorities and the weak which are part of them; there is no civilization, and cannot be, which does not protect these minorities, and which does not adequately help them, drawing the best from its discoveries to help also, and perhaps above all, the needy.
At this point we wonder: ''Multimediality'' or ''Hypermediality'', how do they stand in relation to these minorities and in particular, what is computer science doing today for one of these minorities, struggling with one of the most feared handicaps: blindness? Let us therefore look at ''Multimediality'' in terms of this serious physical disability. It is natural to think that the relative importance of images in this ambit must necessarily decrease enormously in favour of sounds, or words, generated by the computer precisely in order to sustitute images. Given this, the advantages of ''Multimediality'' are totally on the side of those who cannot see but who use a computer.
The important concept to highlight is the following: wherever there is a serious physical disability like blindness, computer science must study and solve only the problem of the ''differences'' to be resolved and not

undertake research for solutions which by their very nature tend to exclude the blind person from whatever new technology which, developed from market forces, succeeds in creating yet another step on which to trip up. In other words, attention must only be given to the tools which help overcome the handicap and the temptation must be resisted to study 'Ad Hoc' solutions, perhaps more profitable or desirable, which for obvious technical-economic reasons would end up excluding the blind person from progress reached. This would be unjust - and thus uncivil - and not just wrong.

There are organizations which have already taken this path, with success which proves the worthiness of the approach just mentioned. And Italy, we are proud to say, is amongst the nations which have better understood this task. There still remains a lot to do, above all if one considers how ''Multimediality'' and ''Hypermediality'', gauge and symbol of the direction in which the computer science of tomorrow is going, tend to put more and more emphasis on images, and therefore by their very nature become more difficult for the blind person to use.

What is to be done?
Simply this: remain faithful to the concept of 'Differences' to resolve, and find other solutions for those who cannot and will never be able to see the marvel of colours in high definition.
In order to better understand what we are talking about it is a good idea to look at what the blind person can do today with his computer.

There are software products, based on unlimited vocabulary voice synthesis, which simply substitute ears for eyes: which thus permit a sighted person and a blind person to use the same hardware and software without noticeable differences (without considering that individual propensity often makes the blind person a more brilliant and quicker user than a sighted person). On this basis, at least in Italy, much has been built: to the first personal computer, learnt with difficulty and used to write and make every day notes, many blind people have added a modem, and have therefore learnt to link up to traditional data banks and to the numerous new ones which are continually springing up in Italy and abroad; they have therefore been able to consult the databases available to VIDEOTEL users; they have then added an optical reader in order to gain access to the most widespread compact discs; they have recently even been able to access the most sophisticated character recognition technology (ICR: Intelligent Characters Reading) so as to directly read vocally books written in black and white, neglecting in some cases the irreplaceable language developed by BRAILLE at the beginning of the last century.

Today in Italy, we are proud to say, the blind person is not discriminated against in any way with respect to the sighted person in the use of personal computers; on the contrary, we can state that this important work tool is for the blind person a visual prosthesis - indispensable in overcoming the visual handicap, and thus also for personal and professional realization just like and better than the sighted person, in all fields of activity and work. Today in Italy the blind person can happily consider abandoning the activities traditionally assigned to him such as switchboard operator.

Let us quote the example of the switchboard operator from Trieste, G. L. UGO who would like to be a writer and who has already written and published his first book based on a historical subject; or of the young and brilliant lawyer from Salerno, S. SALVATI, who with his personal computer is rapidly scaling the heights of a brilliant, muchdeserved legal career; or of A. MARTINELLI, a language student from Milan who is working as a translator.

Therefore what must be done is to resolve the differences in a strict, constant symbiosis with computer science in its continuous and untidy growth, and not to look for 'Ad Hoc' tools which increase humiliation by highlighting deficiency; to smooth out the corners with constant and competant research and not to build cathedrals in the desert; and all this with the vigilant presence of the various organizations for the blind.

It remains to be understood which path must be taken in the future. We have spoken of Voice Synthesis as an initial element of research, meant to resolve the differences imposed by Nature on the blind. 'Hypermediality' leads us though towards the increasingly perfect processing of images. The paths therefore coincide. Just as the increasingly tight fusion between sounds and images decisively shows which path to take in order to rea-

lize 'hypermediality', also the social aspects of computer science dedicated to the blind person must veer towards solutions in which images increasingly embrace sound, or voice: with one importance difference which clearly defines the confines of research: the Voice which accompanies the images must describe the images themselves, besides making comments. It is therefore necessary to study other software tools besides those which have already been developed in the past which succeed in an automatic way in describing to the blind person the images which are shown on the screen together with words, chosen and delivered opportunely even today.

It is a problem of civility and good will. All those who directly or indirectly influence the tumultuous progression towards the future of computer science today; all those who contribute to the formation of the concept of "Multi" or "Hypermediality"; all those working in the field must be aware of this need for civility, and set the path towards it so that blind people can fully enjoy the fruits of their work: without difference, without discrimination and therefore without injustice.

Despite the unjust punishment inflicted on the blind by Nature.

A PROJECT ON SOUNDS IN ARTIFICIAL REALITIES*

Francesco Gardin, Goffredo Haus,
Ilaria Macchia, Gloria Molendi
Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano

RIASSUNTO

Questo articolo descrive un progetto di ricerca, denominato 'SOUNDS WORLD', in fase di realizzazione, che ha per obiettivo l'aggiunta di suoni ad una realtà artificiale, inizialmente una stanza tridimensionale, in modo che la maggior parte degli aspetti cognitivi relativi alla percezione dei suoni nel mondo reale, siano resi possibili nella stanza artificiale. Il modello di propagazione del suono utilizzato è stato volutamente tenuto molto semplice, in maniera da poter essere calcolato in tempo reale, ma si è tentato di mantenere il più possibile le caratteristiche percettive di base del suono in un ambiente reale.
La comunicazione verbale tra due, o più, utenti che condividono la stessa stanza artificiale, ognuno sulla propria stazione di lavoro, pertanto percependo visivamente l'un l'altro nella stanza artificiale, sarà basata sul modello di propagazione dei suoni sopra citato.
Questa è la prima parte di un progetto di dimensioni più vaste che ha come obiettivo la creazione di mondi artificiali complessi dove le persone possono muoversi e parlare tra di loro, come se fossero nel mondo reale.

ABSTRACT

This paper describes an ongoing project called 'SOUNDS WORLD', where the goal is to add sound to an artificial reality, initially a 3-D room, so that most of the cognitive aspects of sound perception in the real world are made possible in the artificial room. The sound propagation model used has been kept extremely simple, so that it can be computed in real time, but the basic perceptual features of sound in real world are maintained as much as possible.
Speech communication among two, or more, users sharing the same artificial room on different individual workstations, perceiving each other visually in the artificial room, will be based on the above mentioned model of sound propagation.
Extensions to the properties of sound as it behaves in the real world will be explored through modifications of the basic proposed sound model.
This is the first part of a larger project which aims at the creation of a complex artificial world where people can move around and talk to each other as if they were in the real world.

1. INTRODUCTION

Artificial realities (GDA 89) are synthetic worlds where users perform operations which differ in manner and goals. Laws of real world come to be simulated, extended or simply changed according to necessity.
The project we are carrying out experiments with the use of sound in an artificial reality. Until today much has been done about spacial features of artificial reality, but little attention has been paid to sound, which is a basic cognitive factor in human perception.
Our aim is to build a simple 3D artificial environment where we can add objects as they become necessary. A simple model of sound propagation has been proposed and its cognitive soundness is being tested.

* This project is partially funded by the Italian National Research Council in the frame of the "MUSIC Goal: Intelligent Music Workstation", Finalized Project "SISTEMI INFORMATICI E CALCOLO PARALLELO", Subproject 7 "SISTEMI DI SUPPORTO AL LAVORO INTELLETTUALE

Association of sound to space makes possible the .... All this will be carried out as a prototype on a Silicon Graphics workstation which is connected with a Macintosh equipped with MIDI devices. The user will be given headphones to perceive sound, a microphone for voice communication and a joystick to move inside the artificial environment.

2. A SIMPLE SOUND PROPAGATION MODEL

The proposed model of sound propagation should capture the properties of sound according to our needs: basic cognitive soundness i.e. the elementary features that we experience and are used to in the real world: so, for instance, if we get closer to a sound source we expect to hear it louder, if we hear a sound we should be able to locate its spatial location, etc.
Although from the theoretical point of view it would be possible to implement a very detailed model of sound propagation and interference, using techniques which are similar to ray-tracing]x], this would be difficult to compute in real time, and might be over-detailed for the minimal cognitive quality that we are aiming at initially.
Moreover, we have chosen a hybrid modelling technique, namely a combination which attempts to minimize the computation required and maximize the quality of the sound generated. The propagation model that we are implementing is based on a simple calculation of the relative distance and orientation, in the artificial room, between the source and the receiver. The expected amplitude of sound is therefore calculated only for the spatial coordinates where the receiver is positioned. Since for the sake of simplicity we constrain our environment to be anechoic, the interference between sounds is not calculated, instead each individual sound is generated as it should be perceived at the receiver position as if it was the only one. Combination of simultaneously generated different sounds takes place in the real world of the user, therefore drastically reducing the computation required.
A sequence of experiments has been planned, with increasing levels of properties to be tested in the proposed sound propagation and interference hybrid model, in order to study its cognitive aspects.

We begin with a still single sound source and one user (the receiver) (Fig. 1).

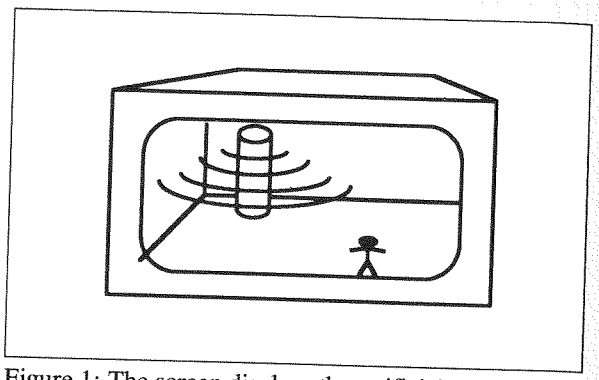


Figure 1: The screen displays the artificial environment from the user viewpoint.

The second level is represented by the addition of a wall between the sound source and the receiver (Fig. 2, Fig. 3).

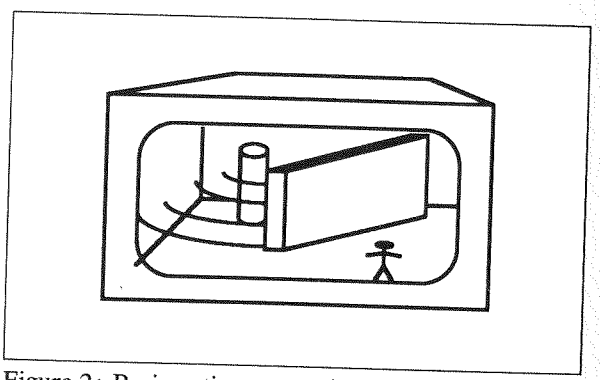


Figure 2: By inserting a sound-wall between the user and the sound source, sound perception should change.

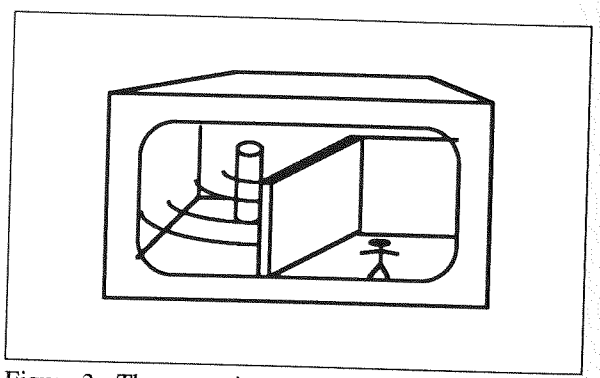


Figure 3: The room is soundproof.

We then add another still source to test interference (Fig. 4) as proposed above in our hybrid model.

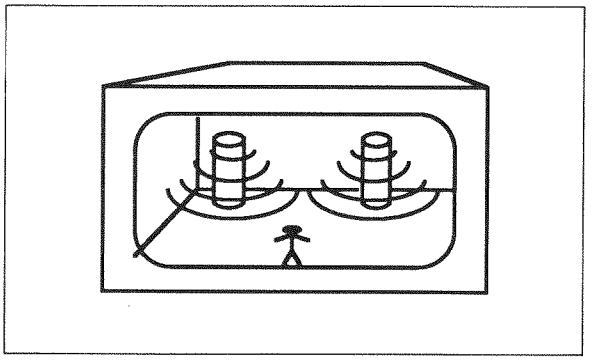


Figure 4: A second sound source has been added.

Then we must extend the model in order to deal with moving sound sources, so that we can study sound interaction among several agents. In order to do this the environment has to be shared: each user will perceive sounds and images of the artificial reality according to his own position as an agent inside the artificial room (Fig. 5). This is made possible by the use of a 3-D interactive real time navigation system developed by [1] and substantially improved by [8], on a Silicon Graphics workstation. The system has been extended to give simultaneous access to two users, so that each one of them is sharing the same artificial room from his own viewpoint, as shown in (Fig. 5).

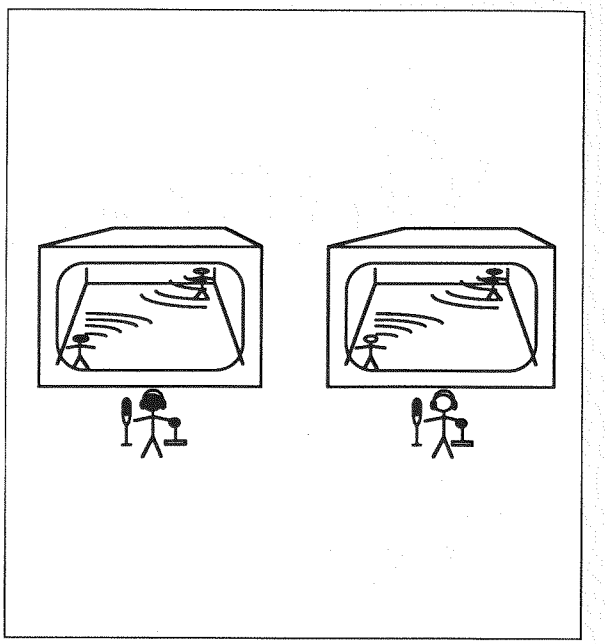


Figure 5: Two agents share the same environment.

The ability to handle sounds and images, together with the possibility of having several users physically in different places, simultaneously sharing the same virtual space, is an interesting metaphor of communication.

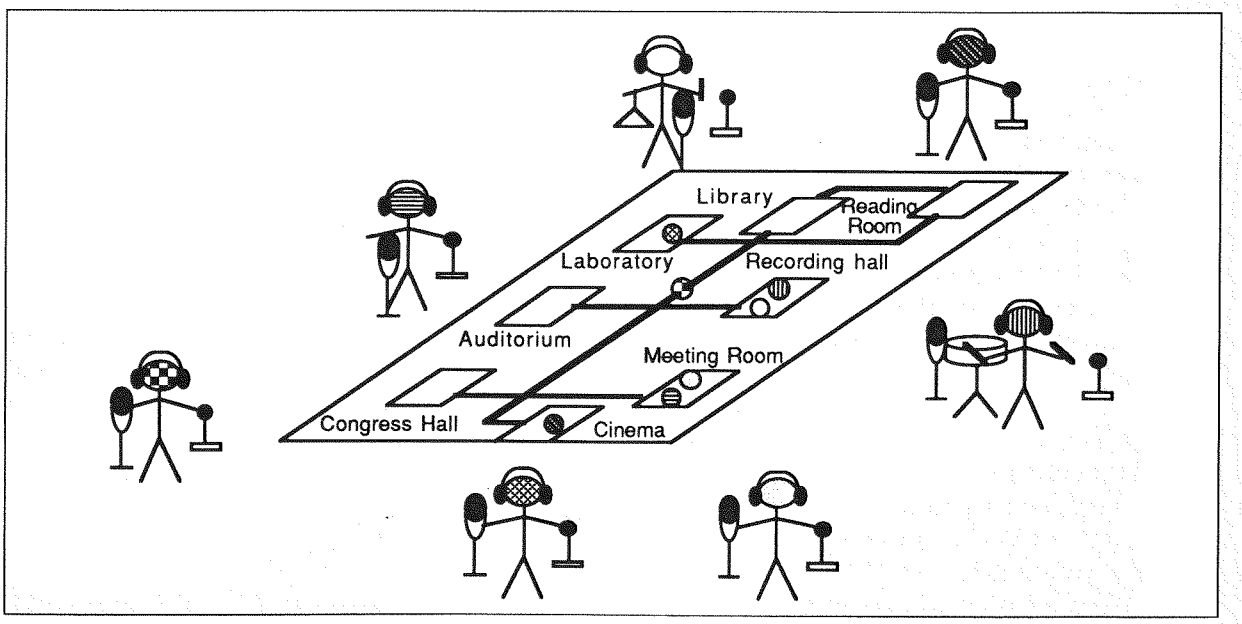


Figure 6: Several agents can interact in the same artificial environment through microphone, headsets and joystic so that they can speak, hear and move around.

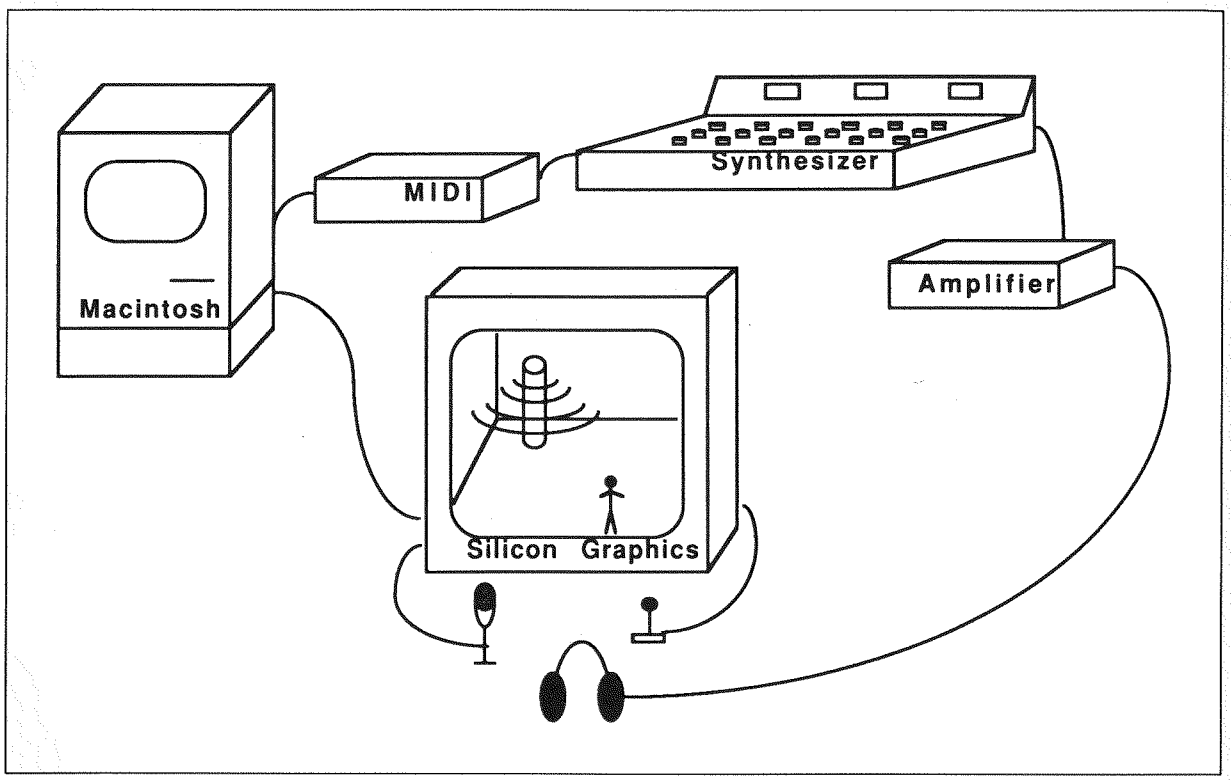


Figure 7: System architecture

Moreover the limits and failures of traditional communicating systems, such as videophone and telelectures, are overcome (Fig. 6).

A fundamental aspect of artificial realities are the cognitive relations connecting the user to the artificial world. The laws governing these connections, given the artificial nature of the environment, can be chosen in an arbitrary way. We therefore believe that such realities might be used as laboratories in the study of behavioural features of Man in communication, perception and cognition (Gardin 90).
Moreover, the basic sound propagation model can be modified, so that additional properties can be added to sound. As an example of what we mean by this we can think of sound waves propagating along contours of objects, so that each time we get close enough to an object we perceive such waves and can follow along the surface of the object the propagation paths. A second example can be represented by the notion of sound tracks, namely sounds that stick to the first object they hit, and are reflected repeatedly only for a very limited distance: a room will store all conversations that took place within it. By changing the properties of the model we can generate familiar objects such as tape recorders, or megaphones, but also completely new objects such as .....

3. SYSTEM ARCHITECTURE

The architecture which has been devised for the system is shown in (Fig. 7). Since a MIDI interface for the Silicon Graphics workstation wasn't available, control has to be transferred to a Macintosh, which in turn is interfaced to the MIDI.
The next step, after the conclusion of the experiment with the current sound propagation model, will probably be developed on a single workstation directly interfaced with the MIDI.

4. CONCLUSIONS

The project described in this article represents a first step towards the creation of realistic artificial realities,

where sound is combined with 3-D images, making interaction possible among users sharing the same environment. Although the proposed sound propagation model is very simple we hope it will capture the essential cognitive features of sound as we experience it in the real world. **As for any component of an artificial reality, sound can be modified in such a way that unique properties are made possible.**
Eventually a network of sound worlds, soon not just confined to rooms but also encompassing buildings and small towns, will represent a first step towards a new form of human communication.

ACKNOWLEDGMENTS

We would like to thank Jaron Lanier from VPL, Gianni Degli Antoni from the Dept. of Computer Science, University of Milan, for their stimulating comments, Daniele Marini, from the Dept. of Computer Science, University of Milan, for the discussion we had on the ray-tracing techniques to model sound, and the support that he gave making available the Silicon Graphics workstations belonging to his Computer Graphics Lab., Flavio Argentesi and Sirio Stramaccia from JRC Ispra for their constant support since the early stages of the 'SOUNDS WORLD' project.
Special thanks are due to the L.I.M.-Laboratorio di Informatica Musicale , Dept. of Computer Science, University of Milan, staff for the scientific and technical support provided.
Artificial Reality Systems srl in Milan, for their help throughout the project.

REFERENCES

[1] C. Chiaravalle, SVILUPPO PER LA PROGETTAZIONE DI UN SISTEMA DI NAVIGAZIONE PER LA CREAZIONE DI AMBIENTI NATURALI E COMPOSIZIONE DI ESSI CON OGGETTI SINTETICI MEDIANTE L'USO DI UN VIDEODISCO, Master Thesis, Dept. of Computer Science, Univ. of Milan, 1988

[2] G. Degli Antoni, FROM REALITY TO HYPERMEDIA THROUGH ARTIFICIAL REALIITY, in UHG89, pp.10-15, 1989

[3] G. Degli Antoni, ARTIFICIAL WORLDS, in UHG89, pp.8-9, 1989

[4] F. Gardin, A MODEL FOR ARTIFICIAL REALITY, in Note di Software 48/49, 1990

[5] F. Gardin, A MODEL FOR ARTIFICIAL REALITY, in Note di Software 48/49, 1990

[6] J. Lanier, PRIVATE CONVERSATION, Redwood City, 1989

[7] W. M. Newman, R.F. Sproull, PRINCIPLES OF INTERACTIVE COMPUTER GRAPHICS, McGraw-Hill Inc., 1981

[8] S. Stramaccia, MANUALE D'USO PER LA CREAZIONE DI AMBIENTI 3-D NAVIGABILI, Rapporto Interno, JRC Ispra, 1989

EDITO DA CALEIDOGRAF S.R.L.
VIA L. DA VINCI N.4/6 TEL. 039 - 587541
22058 OSNAGO (CO)

Spedizione in abbonamento postale, Gruppo IV, 1° semestre.
Piacenza F
Pubblicità Inferiore al 70%
N. 36, 48/49 - Giugno-Ottobre 1990

TAXE PERCUE
TASSA RISCOSSA
P.P. P.T. Piacenza F